Importance sampling for high speed statistical Monte-Carlo simulations

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computer Science

CASA-Report 09-37
November 2009

Importance sampling for high speed statistical Monte-Carlo simulations

By

E.J.W. ter Maten, T.S. Doorn, J.A. Croon, A. Bargagli,
A. Di Bucchianico, O. Wittich

Centre for Analysis, Scientific computing and Applications
Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven, The Netherlands
ISSN: 0926-4507

Technical Note TUE-CASA-2009

Issued: April 1/2009

Importance Sampling for High Speed Statistical Monte-Carlo Simulations

Designing very high yield SRAM for nanometre technologies with high variability

E.J.W. ter Maten, T.S. Doorn, J.A. Croon, A. Bargagli, A. Di Bucchianico, O. Wittich

Authors' address E.J.W. ter Maten Jan.ter.Maten@nxp.com

T.S. Doorn Toby.Doorn@nxp.com

J.A. Croon Jeroen.Croon@nxp.com

A. Bargagli Agnese.Bargagli@nxp.com

A. Di Bucchianico A.D.Bucchianico@tue.nl

O. Wittich O.Wittich@tue.nl

Title: Importance Sampling for High Speed Statistical Monte-Carlo Simulations

Author(s): E.J.W. ter Maten, T.S. Doorn, J.A. Croon, A. Bargagli, A. Di Bucchianico, O. Wittich

Technical Note: TUE-CASA-2009

Additional Numbers:

Subcategory:

Project:

Customer:

Keywords: Importance sampling; Monte Carlo methods; Statistics and discrete mathematics; Numerical mathematics; Rare events; Extreme events; High yield SRAM; MOSFET variability; MOSFET matching

Abstract: As transistor dimensions of Static Random Access Memory (SRAM) become smaller with each new technology generation, they become increasingly susceptible to statistical variations in their parameters. These statistical variations can result in failing memory. SRAM is used as a building block for the construction of large Integrated Circuits (IC). To ensure SRAM does not degrade the yield (fraction of functional devices) of ICs, very low failure probabilities of $P_{\text{fail}} = 10^{-10}$ are strived for. For instance in SRAM memory design one aims to get a 0.1% yield loss for 10Mbit memory, which means that 1 in 10 billion cells fails ($P_{\text{fail}} \leq 10^{-10}$; this corresponds with an occurrence of -6.4σ when dealing with a normal distribution).

To simulate such probabilities, traditional Monte-Carlo simulations are not sufficient and more advanced techniques are required. Importance Sampling is a technique that is relatively easy to implement and provides sufficiently accurate results. Importance sampling is a well known technique in statistics to estimate the occurrences of rare events. Rare or extreme events can be associated with dramatic costs, like in finance or because of reasons of safety in environment (dikes, power plants). Recently this technique also received new attention in circuit design.

Importance sampling tunes Monte Carlo to the area in parameter space from where the rare events are generated. By this a speed up of several orders can be achieved when compared to standard Monte Carlo methods. We describe the underlying mathematics. Experiments reveal the intrinsic power of the method. The efficiency of the method increases when the dimension of the parameter space increases.

The method could be a valuable extension to the statistical capacities of any circuit simulator A Matlab implementation is included in the Appendix.

Conclusions:

A 0.1% yield loss for 10Mbit SRAM memory, which means that 1 in 10 billion cells fails ($P_{\text{fail}} \leq 10^{-10}$) can be efficiently estimated by Monte Carlo methods that are tuned by Importance Sampling. Importance sampling brings Monte Carlo to the area in parameter space from where the rare events are generated. By this a speed up of several orders can be achieved when compared to standard Monte Carlo methods. The efficiency of the method increases when the dimension of the parameter space increases.The method can be efficiently implemented in any circuit simulator and can be extended to allow for adaptive tuning of the rare event density distribution.

A preliminary version of Importance Sampling has been implemented using NXP Semiconductors' circuit simulator Pstar with Matlab post processing and has been demonstrated to work correctly. The method has been applied to estimate the probability distribution of all 4 SRAM cell parameters: Static Noise Margin (SNM), Write Margin (WM), Read Current and Bitline Leakage Current. A good correspondence of Importance Sampling Monte Carlo (ISMC) and traditional Monte Carlo simulation was shown for the relevant probability range.

For the SNM, it is shown that extrapolation of standard MC simulations overestimates the yield. In addition to the benefit of ISMC simulations, it has been shown that extrapolation of the Gaussian distributions of the individual SNM 'eyes' (specific enclosures of two curves) yields results in accurate yield estimation. The results of the latter method are in agreement with ISMC simulations.

The Read Current distribution deviates strongly from a Gaussian distribution and its distribution can therefore not be extrapolated. The use of extrapolated distributions would result in a pessimistic Read Current and could thus lead to over-design of the memory cell and/or memory architecture. Importance Sampling or a technique with similar statistical accuracy is required to make correct decisions in the design process.

The WM can be estimated with extrapolated Gaussian distributions. Although a small difference of the WM at $P_{\text{fail}} = 10^{-10}$ is observed between extrapolated MC and ISMC, this difference is not significant.

To determine the SRAM Total Leakage Currents the average current per cell is multiple by the number of cells in the instance. A guideline is proposed to guarantee that Bitline Leakage Currents do not compromise SRAM functionality.

We introduced Importance Sampling as a technique to efficiently perform failure analysis. To prove benefits over standard Monte Carlo we applied and extended knowledge from Large Deviation theory. The basics of the method can easily be implemented in a circuit simulator or in a shell procedure around a circuit simulator. For a refined procedure, involving adaptive sampling, we introduced a new approach. Here some intial tests were made using 1-dimensional functions. The real benefit must come from problems with parameters in a higher dimensional space. This will require further research.

Apart from the studied Importance Sampling we also described two additional variants (weighted importance sampling, regression importance sampling) and indicated how one may reduce the variance of a particular variant of Importance Sampling by optimizing a parameter.

Contents

Section 1

Importance Sampling: An SRAM Design Perspective

Importance sampling is a well-known technique in statistics to simulate the occurrences of rare events [17] (1964). Rare or extreme events can be associated with dramatic costs, like in finance or because of reasons of safety in environment (dikes, power plants). Recently this technique also received new attention in circuit design. For instance in SRAM memory design one aims to get a 0.1% yield loss for 10Mbit memory, which means that 1 in 10 billion cells fails ($P_{\text{failure}} \leq 10^{-10}$; this corresponds with an occurrence of -6.4σ when dealing with a normal distribution). Importance sampling tunes Monte Carlo to the area in parameter space from where the rare events are generated (corresponding to the tails of the distribution). By this a speed up of several orders can be achieved when compared to standard Monte Carlo methods. We describe the underlying mathematics. Experiments reveal the intrinsic power of the method. The efficiency of the method increases when the dimension of the parameter space increases.
The method would be a valuable extension to the statistical capacities of Pstar [42]. We also describe a global description for an efficient implementation in Pstar. A Matlab implementation is included in the Appendix.

1.1 YIELD AND SRAM YIELD PREDICTION

Static Random Access Memory (SRAM) is one of the main building blocks of any digital integrated circuit (IC). A large digital IC is often referred to as "System on Chip" (SoC), since one SoC consists of a large number of system blocks, including memory. For mobile phone chips, these blocks can include data receivers/transmitters (for GSM, UMTS, Bluetooth, Wifi, etc) and digital video and audio processing. Together, all of these blocks can add up to several 100 million transistors. Each of these transistors has to operate correctly and has to be correctly connected to the rest of the system.

Just one single failing transistor leads to a SoC not being 100% correct, and can prevent it from being sold. The profit a semiconductor company makes is directly related to the fraction of SoC's that are functional after fabrication. Therefore, the probability that a transistor fails has to be very, very small. The fraction of functional chips is commonly referred to as yield. Typically, the yield of a factory has to be above 70%-80%, before it can profitably operate. For good products, the yield is above 90%.

SRAM has a higher probability of not functioning than "normal" digital circuitry, since it is not a purely digital design. The cell is built around a read/write trade-off. It has to be stable

enough to be read without changing its data, yet unstable enough to be written when desired.

Up to half of the chip area of a SoC can be consumed by SRAM. Since this is a large portion of the chip, a lot of effort is put into reducing the size of the memory cells. Reducing the size of the memory cells increases the probability that they fail, because fluctuations in the technology parameters have a larger impact on smaller transistors. Special care is taken to guarantee that SRAM does not limit the SoC yield and functions correctly in the presence of these parameter fluctuations. Currently (45 nm technology), it is assumed that each SoC contains 10 million memory cells (10 Mbit) and that 1 in 1000 SoC's does not function correctly because of the SRAM. This results in a failure probability of the memory cells of $P_{\text{fail}} = 10^{-10}$.

1.2 IMPORTANCE SAMPLING MONTE CARLO SIMULATIONS

To predict SRAM failure rates, a standard Monte Carlo method is currently used [14]. This method uses the physical distributions of the statistical transistor parameters, threshold voltage V_t and (current) amplification factor β, to randomly introduce variations to each transistor. Both V_t and β have a Gaussian distribution. The simulator randomly draws values for V_t and β for each transistor, based on the Gaussian distribution. By definition, using a Gaussian distribution results in most of the trials being drawn from around the mean of the distribution. To estimate extreme probabilities, the tails of the distribution are more important than the average values. Consequently, it is desirable to have more samples drawn from the tail of the distribution. A suitable distribution would be one that has higher probabilities in its tails than a Gaussian distribution. A uniform distribution is one of the simplest examples of such a distribution (Figure 1.1). Using an importance sampling distribution g as an input for Monte-Carlo simulations leads to

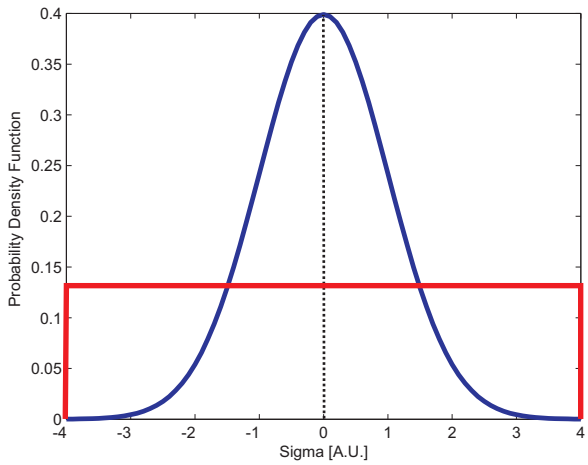

Figure 1.1: A uniform distribution has higher probabilities in its tails than a Gaussian distribution.

a distorted distribution of the output parameter. This has to be corrected with post-processing. Suppose we are doing Monte-Carlo analysis for the Static Noise Margin of SRAM cells. For each trial, the probability has to be calculated that the drawn value of the input parameter (V_t or β) would have occurred in the (original) normal distribution f. This is done by integrating the

distribution function, which for V_t gives:

$$F_{V_t} = \int_{\text{binwidth}} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\left(\frac{V_t - \mu}{\sigma}\right)^2 dV_t\right].$$

Naturally, the original distribution f (so in particular, the parameters μ and σ) has to be known to be able to do this. The binwidth is known from the uniform distribution, as is shown in Figure 1.2. For 1 trial, the binwidth is $V_{t_{\text{range}}}/N$, with N the number of trials. So each trial has to

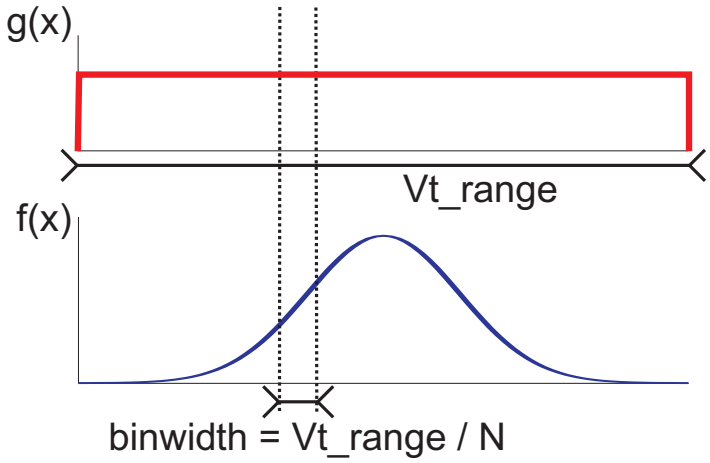

Figure 1.2: The probability that a trial is drawn from the interval binwidth is $g(x)$binwidth. For 1 trial, the binwidth is $V_{t_{\text{range}}}/N$, with N the number of trials.

be corrected with the probability that it would occur if the input parameter (V_t) were normally distributed.

$$P\left(SNM_{trial}\right) = f(V_t)\frac{1}{N}\frac{1}{g\left(V_t\right)} = f\left(V_t\right) \cdot V_{t_{\text{range}}}/N.$$

To make the distribution for SNM, a number of bins has to be defined. The probabilities that a certain combination of V_t's lead to a certain SNM have to be summed.

$$P\left(SNM_{bin}\right) = \sum_{\text{bin}} f\left(V_t\right) \cdot V_{t_{\text{range}}}/N = \frac{1}{N}\sum_{\text{bin}} \frac{f\left(V_t\right)}{g\left(V_t\right)}.$$

A more formal notation uses indicator function I to count the number of occurrences in a bin t and expresses the distribution function $F_{\text{SNM}}(t)$:

$$F_{SNM}\left(t\right) = \frac{1}{N}\sum_{i=1}^{N} I_{\{SNM<t\}} \frac{f\left(V_{t_i}\right)}{g\left(V_{t_i}\right)},$$

where

$$I_{\{SNM<t\}} = \begin{cases} 1 & \text{if } SNM \leq t \\ 0 & \text{else} \end{cases}.$$

1.3 APPLICATION OF IMPORTANCE SAMPLING

Figure 1.3 shows the Static Noise Margin (SNM) density function of an SRAM cell, using a Gaussian sampling distribution (blue) and a uniform sampling distribution (red). Clearly, when

using a uniform density function, the result is a distorted SNM distribution that covers a much wider range than the original distribution. The red distribution in Figures 1.3 and 1.4 has to

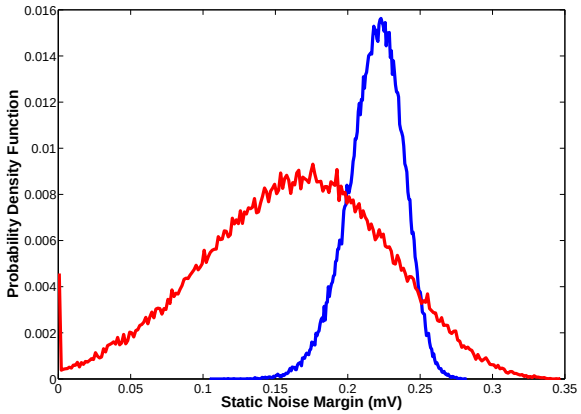

Figure 1.3: Static Noise Margin density function of an SRAM cell using a Gaussian sampling function (blue) and a uniform sampling function (red). The simulations use 50k trials.

be corrected for using a distorted sampling function. Figure 1.4 includes the corrected SNM distribution (green). As is obvious, the corrected density shows much more statistical noise around the mean than the normal SNM density function. This is basically what Importance Sampling does. It trades accuracy around the mean for more accuracy in the tails. Since the tails are more important in this case, this is acceptable behaviour. Figure 1.5 shows the cumulative

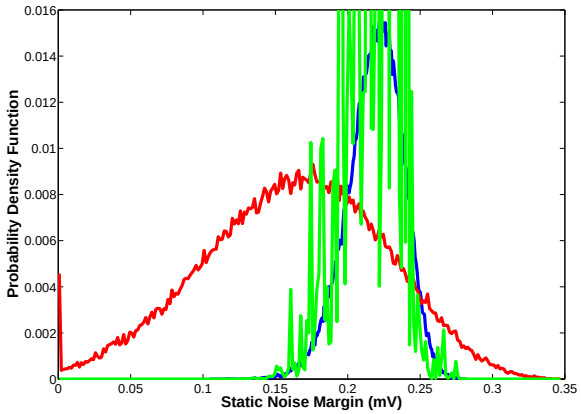

Figure 1.4: SNM density function of an SRAM cell using a Gaussian sampling function (blue), a uniform sampling function (red) and the corrected SNM distribution (green). The simulations use 50k trials.

distribution function of the SNM, using a Gaussian sampling function (blue) and using a uniform sampling function (green). Using Importance Sampling, the distribution extents to much smaller probabilities than without Importance Sampling. The distributions with and without Importance Sampling are on top of each other in the higher probability range (down to approximately $P_{\text{fail}} = 10^{-4}$). This example shows what Importance Sampling costs and what is can bring: increased accuracy in the tails of the distribution at the expense of more noise around the mean.

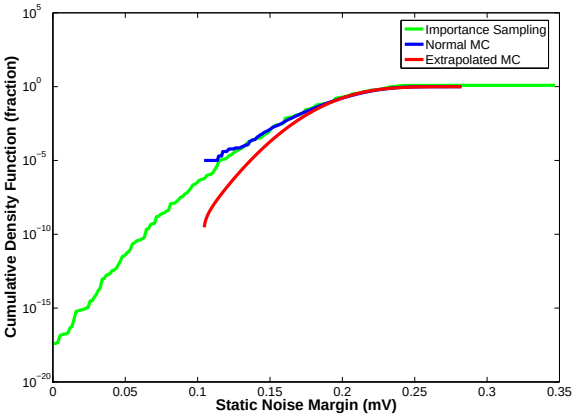

Figure 1.5: SNM density function of an SRAM cell using a Gaussian sampling function (blue), a uniform sampling function (red) and the corrected SNM distribution (green). The simulations use 50k trials. The ”Extrapolated MC” is not discussed here, but will be explained later in the Sections 5 and 6.

Section 2

Basic Statistics and Monte Carlo

Monte Carlo sampling is a well-known method to obtain estimates for probabilistic quantities by simulating appropriate random variables. After a review of basic concepts in probability theory and statistics, this section just summarizes some basic aspects like how many sample points one must take to assure predefined accuracy. Also, for the normal distribution $\mathbf{N}(0,1)$ we relate extreme probabilities to the σ-scale. Finally we will discuss some options for improving Monte Carlo statistics. For specific items related to statistics using the circuit simulator Pstar we refer to [41, 43].

2.1 BASIC PROBABILITY THEORY

In this section we will introduce the basic notions of probability theory. In particular, we will learn about random variables, means, variances, distribution functions, densities, joint distributions, independence and correlation.

Following common usage in statistics, we will denote random variables (theoretical random objects) with capitals and their realizations (actual observed values) with small letters. A (real) random variable X can be seen as a real-valued function that assumes values according to a probability measure p (weighing function)[1]. Probabilities of the occurrence of values[2] are defined as integrals with respect to this measure p:

$$P(a < X < b) = \int_a^b dp(x). \tag{2.1}$$

It is convenient to have a name and notation for probabilities of the form $P(X \leq b)$. The distribution function[3] is the function defined by

$$F(x) = P(-\infty < X \leq x) = P(X \leq x). \tag{2.2}$$

The link between $p(x)$ and $F(x)$ is given by $dp(x) = dF(x)$.
Important properties of a random variable X include the mean or expectation

$$\mathrm{E}(X) = \int_{-\infty}^{\infty} x\, dp(x) \tag{2.3}$$

[1] A mathematically more proper way would be to define an abstract sample space Ω with an abstract measure π on Ω. The random variable is then a map from Ω to the real line and the probability measure p mentioned above is then the induced measure of π on the real line by this map.

[2] The statistical jargon is event.

[3] The full official name is cumulative distribution function.

and the variance

$$\mathrm{Var}(X) = \mathrm{E}\,(X - \mathrm{E}(X))^2 = \int_{-\infty}^{\infty} (x - \mathrm{E}(X))^2 \; dp(x). \tag{2.4}$$

Sometimes it is convenient to expand the square in the definition and write

$$\mathrm{Var}(X) = \mathrm{E}\left(X^2\right) - (\mathrm{E}(X))^2 . \tag{2.5}$$

In case of a discrete-valued random variables these integrals become sums, where the outcomes are weighted with the corresponding probabilities. Another important class of random variables is the class of continuous random variables like the normal and uniform distributions. For such random variables we have the following simplification. The distribution function F has a derivative f, called the density. In terms of this density function f, the above formulas can be explicitly written as

$$P(a < X < b) = \int_a^b dp(x) = \int_a^b dF(x) = \int_a^b f(x)\, dx, \tag{2.6}$$

$$\mathrm{E}(X) = \int_{-\infty}^{\infty} x\, f(x)\, dx, \tag{2.7}$$

$$\mathrm{Var}(X) = \int_{-\infty}^{\infty} (x - \mathrm{E}(X))^2 \; f(x)\, dx = \int_{-\infty}^{\infty} x^2\, f(x)\, dx - \left(\int_{-\infty}^{\infty} x\, f(x)\, dx \right)^2 . \tag{2.8}$$

More generally, if h is an arbitrary function, then one can prove [5] (Chapter 7.2) that

$$\mathrm{E}(\, h(X)) = \int_{-\infty}^{\infty} h(x)\, f(x)\, dx. \tag{2.9}$$

Note that Formulas (2.7) and (2.8) correspond to the special cases $h(x) = x$ and $h(x) = (x - \mu)^2$, respectively, where $\mu = \mathrm{E}(X)$. It is also possible to derive the distribution of a transformed random variable. The easiest way is to work with the distribution function. If h is invertible with inverse h^{-1} and F_X is the distribution function of the random variable X, then the distribution function $F_{h(X)}$ of the transformed random variable $h(X)$ equals

$$F_{h(X)}(x) = P(h(X) \le x) = P(X \le h^{-1}(x)) = F_X(h^{-1}(x)). \tag{2.10}$$

Differentiation of this relation yields the density of the transformed random variable.

We now extend this framework to define the joint distribution of two or more random variables. We illustrate this concept for two random variables X_1 and X_2. The joint distribution function is defined by

$$F_{X_1,X_2}(x_1, x_2) = P(X_1 \le x_1, X_2 \le x_2). \tag{2.11}$$

If we moreover assume that the random variables X_1 and X_2 are continuous, then the joint density is defined as

$$f_{X_1,X_2}(x_1, x_2) = \frac{\partial^2}{\partial u \partial v} F_{X_1,X_2}(u, v)\bigg|_{u=x_1, v=x_2} . \tag{2.12}$$

As a consequence, we have the following generalization of (2.6):

$$P(a_1 \le X_1 \le b_1, a_2 \le X_2 \le b_2) = \int_{a_2}^{b_2} \int_{a_1}^{b_1} f_{X_1,X_2}(x_1, x_2)\, dx_1\, dx_2. \tag{2.13}$$

We may recover the marginal (one-dimension) distribution and density function of X_1 by integrating out X_2 (and vice-versa, of course):

$$\begin{aligned} F_{X_1}(x_1) &= F_{X_1,X_2}(X_1 \leq x_1, -\infty < X_2 < \infty), \\ f_{X_1}(x_1) &= \int_{-\infty}^{x_1} f_{X_1,X_2}(x_1, x_2)\, dx_2. \end{aligned}$$

The above notions allow us to define independence. Two random variables are said to be independent if

$$F_{X_1,X_2}(x_1, x_2) = F_{X_1}(x_1)\, F_{X_2}(x_2) \qquad \text{for all } x_1 \text{ and } x_2 \tag{2.14}$$

or equivalently (if X_1 and X_2 are continuous)

$$f_{X_1,X_2}(x_1, x_2) = f_{X_1}(x_1)\, f_{X_2}(x_2) \qquad \text{for all } x_1 \text{ and } x_2. \tag{2.15}$$

We now define the notion of correlation. We will see that correlation and (in)dependence are closely related, but not exactly the same. In order to define correlation we first need multivariate notion of mean and variance. There is a straightforward multivariate notion of mean. The multivariate generalization of the notion of variance is less straightforward. The covariance of X_1 and X_2 is defined as

$$\mathrm{Cov}(X_1, X_2) = \mathrm{E}\left((X_1 - \mathrm{E}(X_1))\,(X_2 - \mathrm{E}(X_2))\right). \tag{2.16}$$

Note that if $X_1 = X_2$, then (2.16) reduces to (2.4). It is often convenient to scale (2.16) to the interval $[-1, 1]$. The *correlation* between X_1 and X_2 is defined as

$$\mathrm{Cor}(X_1, X_2) = \frac{\mathrm{Cov}(X_1, X_2)}{\sqrt{\mathrm{Var}(X_1)\,\mathrm{Var}(X_2)}}. \tag{2.17}$$

It follows from the Cauchy-Schwarz inequality for the $L2$ integral inner product that the correlation is indeed a number between -1 and 1. The random variables are said to be *uncorrelated* if $\mathrm{Cov}(X_1, X_2) = 0$ or equivalently $\mathrm{Cor}(X_1, X_2) = 0$. It follows from expanding the brackets in (2.16) that yet another equivalent way of expression that X_1 and X_2 are uncorrelated is that $\mathrm{E}(X_1\, X_2) = \mathrm{E}(X_1)\,\mathrm{E}(X_2)$. Writing out the definition of expectations, we easily see that if X_1 and X_2 are independent, then they are also uncorrelated (and hence, if there is non-zero correlation, then there must be dependence). The converse is not true. An easy counterexample is taking X_1 as a zero mean normal variable and $X_2 = X_1^2$. Then an easy calculation shows that $\mathrm{Cov}(X_1, X_2) = 0$, but obviously X_1 and X_2 are dependent. However, if X_1 and X_2 are jointly normally distributed *i.e.*, their joint distribution function is bivariate normal, then a zero correlation implies independence.
There are many well-known classes of probability distributions like the normal distribution and the uniform distribution. We refer to the Appendix for basic facts about the most common probability distributions.

2.2 BASIC STATISTICAL THEORY

In this section we introduce the basic statistical concepts. In particular, we will discuss estimators, parameters, unbiasedness, efficiency.

The previous section described the basic probabilistic framework. We now turn to the statistical side of it. In practice one often uses classes of probability distributions like the normal distributions. Such classes depend on one or more parameters. It is the task of statistics to choose

and validate choice of classes of probability models and given such a choice, to extract as good as possible information on these parameters from data. Assume N independent identically distributed observations X_k, $k = 1, \ldots, N$ (such a set of random variables is called sample in statistics). We denote their common mean and variance by μ and σ^2, respectively. The (sample) mean $\widehat{\mu}_N$[4] is defined by

$$\widehat{\mu}_N = \frac{1}{N} \sum_{k=1}^{N} X_k. \tag{2.18}$$

It is sometimes useful to compute the sample mean sequentially according to the recursive formula

$$\widehat{\mu}_N = \frac{1}{N} \left((N-1)\widehat{\mu}_{N-1} + X_N \right). \tag{2.19}$$

Since $\widehat{\mu}_N$ depends on the random sample $X_1, \ldots, X_N$, it is a random variable too. A random variable (or random vector in a multidimensional setting) like $\widehat{\mu}_N$ that is constructed to get an idea of a theoretical, unknown parameter (here μ) is called an estimator. The observed value of an estimator is called estimate (hence, an estimate is a number or in a multidimensional setting a vector). Note the difference between the daily use of the verb estimate and the statistical use here. For any sample from a distribution with a finite mean, the estimator is always (*i.e*,, not depending on the actual probability law of the X_i's as long as all expectations are finite) *unbiased*, meaning,

$$\mathrm{E}\left(\widehat{\mu}_N\right) = \mathrm{E}\left(\frac{1}{N} \sum_{k=1}^{N} X_k \right) = \frac{1}{N} \sum_{k=1}^{N} \mathrm{E}\left(X_k\right) = \frac{1}{N} N\mu = \mu. \tag{2.20}$$

The expectation shows whether there is a systematic deviation from the true, unknown mean. In order to assess the accuracy (fluctuations) of an estimator, we need to consider the variance too

$$\mathrm{Var}\left(\widehat{\mu}_N\right) = \mathrm{Var}\left(\frac{1}{N} \sum_{k=1}^{N} X_k \right) = \frac{1}{N^2} \sum_{k=1}^{N} \mathrm{Var}\left(X_k\right) = \frac{1}{N^2} N\sigma^2 = \frac{\sigma^2}{N}. \tag{2.21}$$

The ideal estimator is unbiased (expectation equal to the target parameter) with minimal variance. The statistical literature yields results to derive estimators and to check whether they have minimal variance (Cramér-Rao Lower Bound, see *e*.g, [3, Chapter 9]).

If Y is a random variable, then expansion of brackets in the definition $\mathrm{Var}(Y) = \mathrm{E}(Y - \mathrm{E}(Y))^2$ yields that $\mathrm{E}(Y^2) = \mathrm{Var}(Y) + (\mathrm{E}(Y))^2 = \sigma^2 + \mu^2$. Hence,

$$\begin{aligned} \mathrm{E}\left(\sum_{k=1}^{N} (X_k - \widehat{\mu}_N)^2 \right) = \mathrm{E}\left(\sum_{k=1}^{N} \left(X_k^2 - 2\widehat{\mu}_N X_k + \widehat{\mu}_N^2 \right) \right) &= \mathrm{E}\left(\sum_{k=1}^{N} X_k^2 - N\,\widehat{\mu}_N^2 \right) \\ = \sum_{k=1}^{N} \left(\mu^2 + \sigma^2 \right) - N\left(\mu^2 + \frac{\sigma^2}{N} \right) &= (N-1)\sigma^2. \end{aligned} \tag{2.22}$$

We now introduce the *sample variance* $\widehat{\sigma_N^2}$ as estimator for the variance σ^2 (in the statistical literature the sample variance is usually denoted by S^2)

$$\widehat{\sigma_N^2} = \frac{1}{N-1} \sum_{k=1}^{N} \left(X_k - \widehat{\mu}_N \right)^2. \tag{2.23}$$

[4] In statistics, this estimator is usually denoted as $\overline{X}_N$. The common usage is to use Greek letters only for theoretical, true quantities like the mean and variance.

The use of $N-1$ instead of N is explained by the following consequence of (2.22)

$$\mathrm{E}\left(\widehat{\sigma_N^2}\right) = \frac{1}{N-1}\,\mathrm{E}\left(\sum_{k=1}^{N}(X_k-\widehat{\mu}_N)^2\right) = \frac{1}{N-1}\,(N-1)\,\sigma^2 = \sigma^2. \tag{2.24}$$

Clearly $\widehat{\sigma_N^2}$ is unbiased. Note that unbiasedness of the sample variance does not hold for its square root, the sample standard deviation. In general $\mathrm{E}\,(\widehat{\sigma_N}) \neq \sigma$. From the recursion (2.19) we observe that

$$\begin{aligned}(N-1)(\hat{\mu}_N-\hat{\mu}_{N-1}) &= X_N-\hat{\mu}_N, &(2.25)\\ N(\hat{\mu}_N-\hat{\mu}_{N-1}) &= X_N-\hat{\mu}_{N-1}. &(2.26)\end{aligned}$$

With this we obtain a practical recursive formula for $\widehat{\sigma_N^2}$, which can be viewed as a parallel to the recusrion for mean values (2.19)

$$\widehat{\sigma_N^2} \;=\; \frac{N-2}{N-1}\;\widehat{\sigma_{N-1}^2} + \frac{(X_N-\hat{\mu}_N)(X_N-\hat{\mu}_{N-1})}{N-1}\,. \tag{2.27}$$

All formulas presented so far are valid for arbitrary distributions as long as all integrals are finite. In case a distribution for the sample $X_1,\ldots,X_N$ is known, then one may obtain more specific results. For example, if the sample is from a normal distribution with mean μ and variance σ^2, then the sample mean is again normally distributed with mean μ and variance σ^2/N (cf. (2.20) and (2.21)), while the sample variance $(N-1)\widehat{\sigma_N^2}/\sigma^2$ has a χ-squared distribution with $N-1$ degrees of freedom. This yields the extra information

$$\mathrm{Var}\left(\widehat{\sigma_N^2}\right) = \frac{\sigma^4}{(N-1)^2}\,\mathrm{Var}\left(\frac{(N-1)\widehat{\sigma_N^2}}{\sigma^2}\right) = \frac{\sigma^4}{(N-1)^2}\,2\,(N-1) = \frac{2\sigma^4}{N-1}\,.$$

One may also prove under normality that $\mathrm{E}\,(\widehat{\sigma_N})$ is a constant times σ, where the constant depends on the sample size N but not on the mean μ.

There are several ways to check whether a sample follows a given class of probability distributions. There are graphical checks like quantile-quantile plots (for normal distributions, this is often called the normal probability plot), but also so-called goodness-of-fit tests. For the normal distributions, there are dedicated tests like the Shapiro-Wilks test (see *e*.g., [18, Section 7.2.1.3]).

We now present an example of an estimator for a parameter which is not related to means and variances. Here we sample from a uniform distribution on an interval $[0,\theta]$, where the right-end of the interval is unknown and must be estimated from samples $X_1,\ldots,X_N$. An obvious estimator here is $\widehat{\Theta} := \max(X_1,\ldots,X_N)$. It follows from independence that

$$F_{\widehat{\Theta}}(x) = P(\max(X_1,\ldots,X_N)\le x) = P(X_1\le x)\ldots P(X_N\le x) = (x/\theta)^N \quad \text{for } 0\le x\le\theta.$$

Hence, $f_{\widehat{\Theta}}(x) = F'_{\widehat{\Theta}}(x) = N\left(\frac{x}{\theta}\right)^{N-1}\frac{1}{\theta}$ for $0\le x\le\theta$ and thus

$$\mathrm{E}\left(\widehat{\Theta}\right) = \int_0^\theta x\,f_{\widehat{\Theta}}(x)\,dx = \int_0^\theta x\,N\left(\frac{x}{\theta}\right)^{N-1}\frac{1}{\theta}\,dx = \int_0^\theta N\,\left(\frac{x}{\theta}\right)^N dx = \frac{N}{N+1}\,\theta.$$

It now immediately follows that $\widetilde{\Theta} := \frac{N+1}{N}\,\widehat{\Theta}$ is an unbiased estimator for θ

$$\mathrm{E}\left(\widetilde{\Theta}\right) = \frac{N+1}{N}\,\mathrm{E}\left(\widehat{\Theta}\right) = \frac{N+1}{N}\,\frac{N}{N+1}\,\theta = \theta.$$

It is not surprising to see that $\max(X_1, \ldots, X_N)$ is systematically underestimating θ, but it is surprising that there is a factor depending on the sample size N only that may be used to compensate for this.

There is a huge amount of literature on estimation theory. We only briefly mention that there are systematic approaches for developing estimators (Maximum Likelihood, moment methods, entropy methods). Maximum Likelihood is popular because it is asymptotically optimal in the sense that asymptotically ML estimators are unbiased and have minimum variance. There are also methods to investigate for finite sample sizes whether an unbiased estimator has minimal variance (Cramér-Rao lower bound for variances, see [3, Chapter 9], Lehmann-Scheffé theorem, see [3, Chapter 10]).

We conclude this section with an example of an estimator for a function rather than a parameter. The empirical distribution function of a sample $X_1, \ldots, X_N$ is the estimator

$$F_N(x) = \frac{1}{N} \{\#i \mid X_i \leq x\} = \frac{1}{N} \sum_{i=1}^{N} I_{\{X_i \leq x\}}. \tag{2.28}$$

In fact, the empirical distribution function is the distribution function of the discrete probability distribution (see also Appendix D) with masses $1/N$ at the points $x_1, \ldots, x_N$. It is easy to see that $N\,F_N(x) \sim \text{Bin}(N, F(x))$ (a binomial distribution with N trials and success probability $F(x)$), from which it directly follows that $F_N(x)$ is an unbiased estimator for $F(x)$ for any distribution function F. The famous Glivenko-Cantelli Theorem shows that the empirical distribution function uniformly converges to the true distribution function F of the $X_1, \ldots, X_N$ (NB: $\sup$ is a generalized form of $\max$):

$$\lim_{N \to \infty} \sup_{x \in \mathbb{R}} |F_N(x) - F(x)| = 0.$$

The empirical distribution function is implicitly playing an important role in Monte Carlo simulations. MATLAB offers the procedures `cdfplot` and `ecdf` to plot and compute the empirical distribution function.

2.3 GENERATION OF RANDOM VARIABLES

In this section we briefly describe how to generate non-uniform random variables. In particular, we discuss the Inverse Transform Method and Acceptance-Rejection Method.

In a Monte Carlo simulation the values X_k must be randomly chosen according to some distribution density function f. There are several general approaches to achieve this.

For continuous distributions the distribution function $F(x) = \int_{-\infty}^{x} f(u)\,du$ is strictly increasing and thus invertible. In practice one starts with a sample $Y_1, \ldots, Y_N$ taken uniformly from [0,1]. By setting $X_k := F^{-1}(Y_k)$ one obtains $P(X_k < x) = P(F^{-1}(Y_k) < x) = P(Y_k < F(x)) = F(x)$. Thus the X_k are chosen according to the density function f. This procedure (usually referred to as the Inverse Transform Method) works well if there is a closed-form expression for the distribution function F (like for the exponential distribution with distribution function $1 - \exp(-x/\theta)$. Numerical inversion should be avoided, since we then have no control on the obtained distribution function. Especially when one is interested in far-away tails like in SRAM simulation, it is very dangerous to use numerically inverted distribution functions. For a normal density function $f \equiv \text{N}(\mu, \sigma)$, $f(x)$ is defined by

$$f(x) \;=\; \frac{1}{\sqrt{2\pi}\sigma}\, e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}\ . \tag{2.29}$$

Generating a normal density function $f(x)$ can be done by an efficient version of the Box-Muller method [7] (see also [6] and [18]). The monographs [1], [13] and [46] are good general sources for simulation from the statistical point of view. The monograph [6] is a thorough treatment of simulation geared towards using large deviation theory for rare event simulation.

A nice method is the *Acceptance-Rejection Method*. It assumes knowledge of a majorant function[5] q with $q(z) \geq p(z)$ and known integral value $I = \int q(z)\,dz$. Then $\widetilde{q}(z) = \frac{q(z)}{I}$ is also a probability density function. The procedure takes each time two random values Z and Y with Z according to $\widetilde{q}$ and Y according to a uniform density on $[0, 1]$. Then

$$\begin{cases} \text{accept} \quad z \quad \text{if} \quad 0 < y < \frac{p(z)}{q(z)} \\ \text{reject} \quad z \quad \text{if} \quad 0 < \frac{p(z)}{q(z)} < y < 1\ . \end{cases} \tag{2.30}$$

When z is accepted take $X_k = z$, otherwise repeat the procedure. This method is very general and applies to many distributions (especially distributions with bounded densities and finite support), but the drawback is that it may not be very efficient.

A good library for random generators is provided by [50] which is based on [36]. For further reference see also [35]. The StatLib website (`http://www.lib.stat.cmu.edu`) also provides many algorithms.

2.4 MONTE CARLO SIMULATION

In this section we show the basics of Monte Carlo simulation. We show how to obtain estimates of probabilities by generating random variables. We discuss ways to determine the minimally required number of simulation runs, in particular using the Central Limit Theorem, Chebyshev's inequality and Large Deviation Theory.

In this section we explain the basic limit theorems in probability theory that make Monte Carlo simulations work. The aim with Monte Carlo is to take samples $X_1, \ldots, X_N$ and to estimate $\hat{\mu}$ and $\hat{\sigma}$. A basic question then is how accurate these estimations are. Also by checking if for a set A $X_k \in A$ one can estimate $P(A)$. When $A = (-\infty, x)$ one estimates $P(X < x)$.
Assume that $X_1, \ldots, X_N$ are independent, identically distributed random variables with finite mean μ and variance σ^2. This setup is more general than it looks at first sight. Of course, sampling from a given well-known probability distribution is included (*e.g.*, the uniform distribution on an interval). The setup also allows probabilities of events by choosing

$$X_i = I_{\{Y_i \in A\}} = \begin{cases} 1 & \text{if } Y_i \in A \\ 0 & \text{if } Y_i \notin A. \end{cases}$$

for given samples $Y_1, \ldots, Y_N$ from a given probability distribution and a set A (*e.g.*, the set A may be a one-sided interval $(-\infty, t)$ or the complement of a specification interval (LSL, USL)[6]). Random variables like X_i are called indicator random variables and they have a Bernouilli distribution (see also Appendix D). The mean of these last indicator random variables is the probability $P(A) = P(Y \in A)$. The Central Limit Theorem says that the standardized sum X_i converges in distribution to a standard normal distribution, *i.e.*,

$$\lim_{N\to\infty} P\left(\frac{\sum_{i=1}^{N} X_i - N\mu}{\sigma\sqrt{N}} \leq x\right) = \lim_{N\to\infty} P\left(\frac{\widehat{\mu}_N - \mu}{\sigma/\sqrt{N}} \leq x\right) = \Phi(x), \tag{2.31}$$

[5] A majorant function g of a function h has values such that $g(x) \geq h(x)$.
[6] Lower Specification Limit and Upper Specification Limit, respectively.

α	10^{-12}	10^{-11}	10^{-10}	10^{-9}	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
z_α	7.03	6.71	6.36	6.00	5.61	5.20	4.75	4.26	3.72	3.09	2.33	1.28

Table 2.1: Typical values of quantiles of the standard normal distribution (σ-scale).

where $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy$ is the cumulative distribution function of the standard normal distribution, *e.g.*, the normal distribution with mean 0 and variance 1. In fact, this theorem holds under much weaker conditions, but this is usually not important when performing simulations. Note that Φ is monotonically increasing and that, because of the symmetry of $\Phi(x)$ around 0, we have $\Phi(-x) = 1 - \Phi(x)$.

The Central Limit Theorem yields that we may use the following approximative confidence interval for μ. Let Z be a standard normal variable. In the sequel we will assume that $\alpha < 1/2$. We define z_α to be the unique number such that $P(Z > z_\alpha) = 1 - \Phi(z_\alpha) = \alpha$. Note that $z_\alpha > 0$ and $P(|Z| > z_\alpha) = 2\Phi(z_\alpha) = 2\alpha$. To give a feeling for the values that z_α assumes for typical values of α, we provide indicative values in Table 2.4. Combining this notation with (2.31), we obtain

$$\begin{aligned} \lim_{N\to\infty} P\left(-z_{\alpha/2}\frac{\sigma}{\sqrt{N}} < \widehat{\mu}_N - \mu < z_{\alpha/2}\frac{\sigma}{\sqrt{N}}\right) &= \lim_{N\to\infty} P\left(-z_{\alpha/2} < \frac{\widehat{\mu}_N - \mu}{\sigma/\sqrt{N}} < z_{\alpha/2}\right) \\ &= \lim_{N\to\infty} P\left(-z_{\alpha/2} < Z < z_{\alpha/2}\right) = 1 - \alpha. \end{aligned}$$

If we wish to estimate μ within absolute accuracy ε with $100(1-\alpha)\%$ confidence, then $N \geq z_{\alpha/2}^2 \sigma^2/\varepsilon^2$. This result is not useful in practice, since we usually do not know σ. Although (2.31) also holds with σ replaced by $\widehat{\sigma}_N$ (this is not trivial, it requires Slutsky's Lemma ([3, Section 7.7])), this only helps a posteriori.

In the special case of indicator random variables, we may use the following approach. Here we exploit an explicit expression for σ. We write $p = P(X \in A)$ and define

$$\widehat{p} = \frac{\sum_{i=1}^{N} X_i}{N} = \frac{\#\{i \mid Y_i \in A\}}{N}. \tag{2.32}$$

Note that $N\widehat{p} \sim \text{Bin}(N, p)$, *i*.e., $N\widehat{p}$ is binomially distributed with N trials and success probability p (see Appendix D). Hence,

$$E(\widehat{p}) = \frac{1}{N} Np = p, \tag{2.33}$$

$$\text{Stand. dev}(\widehat{p}) = \frac{1}{N}\sqrt{Np(1-p)} = \sqrt{\frac{p(1-p)}{N}}. \tag{2.34}$$

Since a binomial distribution is a sum of independent Bernoulli random variables, the Central Limit Theorem yields

$$P(|\widehat{p} - p| > \varepsilon) = P\left(\frac{|\widehat{p} - p|}{\sqrt{\frac{p(1-p)}{N}}} > y\right) \approx 2\Phi\left(-y\right), \tag{2.35}$$

where $y = \varepsilon/\sqrt{\frac{p(1-p)}{N}}$. We need to solve N from the inequality $2\Phi\left(-y\right) \leq \alpha$, where ε and α determine the required accuracy. Using the definition of z_α from above, we obtain $N \geq$

$p\,(1-p)\,\left(\frac{z_{\alpha/2}}{\varepsilon}\right)^2$. This lower bound for N cannot be used directly, since we do not know p. Before we suggest three approaches to overcome this problem, let us look at a simple numerical example to get a feeling for the order of magnitude of N in relation to p and α. For $\alpha = 0.05$, we have $z_{\alpha/2} \approx 2$. We consider the following cases for $\varepsilon = \nu p$, where $\nu = 0.1$.

1. If there is an "intelligent guess" p^* for p (order of magnitude), then use

$$N \geq p^*\,(1-p^*)\,\left(\frac{z_{\alpha/2}}{\varepsilon}\right)^2 = \frac{1-p^*}{p}\,\left(\frac{z_{\alpha/2}}{\nu}\right)^2. \tag{2.36}$$

   We get that $N \geq \frac{4}{0.01}\,\frac{1-p}{p}$. For $p = 10^{-10}$, we have $N \geq 4\,.10^{12}$.

2. If there is no "intelligent guess" for p, then use worst-case $p = 1/2$ (note that because of symmetry, $p(1-p)$ is maximal for $p = 1/2$), so

$$N \geq \frac{1}{4}\,\left(\frac{z_{\alpha/2}}{\varepsilon}\right)^2 = \frac{1}{4}\,\left(\frac{z_{\alpha/2}}{\nu p}\right)^2. \tag{2.37}$$

   For the extreme example above, this would yield $N \geq 10^{22}$.

3. If the above returns a value of N for which the Central Limit Theorem does not apply, use Chebyshev's inequality (2.38) for suitable U. This inequality is valid for any random variable U with finite mean μ and variance σ^2:

$$P\,(\mid U-\mu \mid > \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2}. \tag{2.38}$$

   Proof of (2.38):

$$\begin{aligned}\sigma^2 &= \mathrm{Var}(U) = \int_{\mathbb{R}} (u-\mu)^2 dp(x) \\ &\geq \int_{\{|u-\mu|>\varepsilon\}} (u-\mu)^2 dp(x) \geq \varepsilon^2 \int_{\{|u-\mu|>\varepsilon\}} dp(x) = \varepsilon^2\,P(|u-\mu|>\varepsilon).\Box\end{aligned}$$

   However, the Chebyshev inequality is very conservative. It may easily yield unnecessarily large values of N. We take $U = \sum_{i=1}^{N} X_i/N$ with X_i Bernouillily distributed with success change p^*. Hence we have $\mu(X_i) = p^*$, $\sigma^2(X_i) = p^*(1-p^*)$ and $\mu(U) = p^*$ and $\sigma^2(U) = \frac{1}{N}p^*(1-p^*)$. Requiring $P\,(\mid U-\mu \mid > \varepsilon) < \alpha$, we obtain

$$\frac{\sigma^2(U)}{\varepsilon} \leq \alpha \quad \Leftrightarrow \quad \frac{p^*(1-p^*)}{N\varepsilon^2} \quad \Leftrightarrow N \geq \frac{p^*\,(1-p^*)}{\alpha\,\varepsilon^2} \quad \text{or} \quad N \geq \frac{1}{4\,\alpha\,\varepsilon^2} = \frac{1}{4\,\alpha\,\nu^2\,p^2}. \tag{2.39}$$

   In our extreme example above, it requires that $N \geq 10^{24}$.

In all cases above, because of the required relative accuracy $\varepsilon = \nu p$, we see that $N \to \infty$ when $p \downarrow 0$.

We now present a method to obtain more refined bounds. This method is based on the so-called Large Deviations theory developed by Cramér and refined by, among others, Ellis and Gärtner (see [6] for a detailed exposition). We will apply this method in Section 3 as well.

Let $\varepsilon = \nu p > 0$ be the wanted *accuracy*, for example if we want our approximation to coincide with p in the first four *relevant* digits, we have to consider the probability of failure

$$P\left(\left|\frac{1}{N}\sum_{k=1}^{N} X_k - p\right| > \nu p\right),$$

where we may choose $\nu \sim 10^{-4}$ or any other typical value.

As a short summary of the Large Deviations theory: Let P_N the probability distribution of the random variable

$$Z_N := \frac{1}{N}\sum_{k=1}^{N} X_k,$$

where the X_k are independent indicater random variables that test whether X_k is in some set A. Thus the X_k have a Bernouilli distribution with success change p. Then $P_N = P_N(A)$ and $\mu(P_N) = p$.
The sequence of these probability distributions P_N satisfy a **Large-Deviation Principle** meaning that there is some *'rate function'* $I : \mathbb{R} \to \mathbb{R} \cup \{-\infty, +\infty\}$ such that

(i) $\limsup_{N\to\infty} \frac{1}{N} \ln P_N(C) \leq -\inf_{x\in C} I(x)$ for all closed subsets $C \subset \mathbb{R}$,

(ii) $\liminf_{N\to\infty} \frac{1}{N} \ln P_N(G) \geq -\inf_{x\in G} I(x)$ for all open subsets $G \subset \mathbb{R}$.

The name rate function will be explained later, after (2.44).
Let X be a Bernouilli variable with success change p. The *logarithmic moment generating function* for X (see Appendix D) is given by

$$\ln\left(\mathrm{E}\left[e^{\lambda X}\right]\right) = \ln\left(q + e^{\lambda} p\right),$$

where as usual $q = 1 - p$. We define the following function

$$\begin{aligned} J(x,\lambda) &= \lambda x - \ln\left(\mathrm{E}\left[e^{\lambda X}\right]\right) && (2.40)\\ &= \lambda x - \ln(q + e^{\lambda} p) && (2.41)\end{aligned}$$

where $x, \lambda \in \mathbb{R}$. We note that an optimum value λ^* must satisfy

$$\begin{aligned} \frac{\partial J}{\partial \lambda} &= x - \frac{p e^{\lambda^*}}{q + p e^{\lambda^*}} = 0, \quad \text{hence}\\ \lambda^* &= \ln(\frac{qx}{p(1-x)}), \quad \text{and}\\ p e^{\lambda^*} &= \frac{qx}{1-x}, \quad \text{and} \quad q + p e^{\lambda^*} = \frac{q}{1-x}. \end{aligned} \tag{2.42}$$

In our case, the rate function can be shown to be equal to

$$I(x) = \sup_{\lambda\in\mathbb{R}} J(x,\lambda) = J(x,\lambda^*) = x \ln\left(\frac{qx}{p(1-x)}\right) - \ln\left(\frac{q}{1-x}\right), \tag{2.43}$$

a function which is continuous on the interval $(0,1)$. Assuming now that $C = [p - \nu p, p + \nu p] \subset (0,1)$ we take $G = \mathbb{R} \setminus C$, we obtain from the Large-Deviation principle above that

$$\lim_{N\to\infty} \frac{1}{N} \ln P\left(\left|\frac{1}{N}\sum_{k=1}^{N} X_k - p\right| \geq \nu p\right) = -\inf_{|x-p|\geq \nu p} I(x).$$

From the identities

$$\begin{aligned} I(x) &= x\ln\left(\frac{q}{p}\right) - \ln q + x\ln x + (1-x)\ln(1-x), \\ I'(x) &= \ln\left(\frac{q}{p}\right) + \ln x - \ln(1-x), \\ I''(x) &= \frac{1}{x(1-x)}, \end{aligned}$$

we see that $I''(x) > 0$, which implies sthat I' is increasing and that I is *convex*. Also $I(0^+) = -\ln(q) > 0$ and $I(1^-) = \ln(q/p) \in \mathbb{R}$. Clearly I can be extended continuously at both $x = 0$ and $x = 1$. Furthermore $I(p) = 0$ and $I'(p) = 0$. Hence $I(p) = 0$ is a global minimum. This implies that actually the infimum of I on $\{x : |x - p| > \nu p\}$ is assumed on the boundary of the interval $[p - \nu p, p + \nu p]$ (see [6, Appendix A]), hence

$$\inf_{|x-p|>\nu p} I(x) = \min\{I(p - \nu p), I(p + \nu p)\}.$$

On the other hand, simplifying this a bit using Taylor expansion

$$I(p + h) = I(p) + hI'(p) + \frac{1}{2}I''(p)h^2 + O(h^3),$$

which is feasible since ν is rather small. Here we note that

$$I''(p) = \frac{1}{p(1-p)} = \frac{1}{pq} = \frac{N}{\text{Var}(Z_N)}.$$

We obtain

$$\begin{aligned} I(p - \nu p) &= \frac{Np^2}{2\text{Var}(Z_N)}\nu^2 + O(\nu^3) = \frac{p}{2q}\nu^2 + O(\nu^3), \\ I(p + \nu p) &= \frac{Np^2}{2\text{Var}(Z_N)}\nu^2 + O(\nu^3) = \frac{p}{2q}\nu^2 + O(\nu^3). \end{aligned}$$

Thus from part (i) of the Large Deviation Principle, we obtain the so-called binomial case of the Cramér bound:

$$P\left(\left|\frac{1}{N}\sum_{k=1}^{N} X_k - p\right| \geq \nu p\right) \leq e^{-N \inf_{|x-p|>\nu p} I(x)} \approx e^{-\frac{N^2p^2}{2\text{Var}(Z_N)}\nu^2} = e^{-\frac{Np}{2q}\nu^2}, \qquad (2.44)$$

for all N with a possible exception of finitely many.

Remark 1. *The upperbound in (2.44) decreases exponentially with N which is better than the one obtained by the Chebyshev inequality (2.38)*

$$e^{-\frac{Np}{2q}\nu^2} \leq \alpha \implies N \geq \frac{2q}{p\nu^2}\ln(\frac{1}{\alpha}). \qquad (2.45)$$

However, one still has that N is large for small values of p. If for instance $p \sim 10^{-3}$ and $\nu = 10^{-4}$, then $q \sim 1$ and the exponent is $\sim -10^{-11}N/2$. For the extreme case discussed above we have $\nu = 0.1$, $p = 10^{-10}$ and an upper bound 0.05 for the probability in (2.44). The Large Deviation principle yields that $N \geq 2\,10^{12}\ln(1/0.05) \approx 6\,10^{12}$, which is quite close to the value if we would have an intelligent choice for the unknown p (cf. page 14).

Remark 2. *Part (ii) from the Large Deviation principle states that the exponential bound for the probability in (2.45) is also valid from below. Thus, in that sense, the bound is sharp and convergence can not be faster than with the speed given above. Note that the Large Deviation estimate does not take into account fluctuations of logarithmic order* $o(1/N)$.

Remark 3. *The conclusion is that Monte Carlo needs* $\sim \frac{1}{p}$ *simulations to obtain an estimate with a guaranteed relative accuracy* ν*. We also see that an extra* k*-th decimal in* ν *increases* N *with a factor* k^2.

2.5 IMPROVEMENT OF RESULTS

In this section we describe some ways to improve the basic Monte Carlo Method as described in the previous section. We briefly discuss antithetic variables, control variates, matching moment technique, and stratification.

The Monte Carlo Method of the previous section is the basic form (sometimes called crude Monte Carlo). There are several general techniques to improve this method. It depends on the case at hand, whether a given improvement technique can be implemented and is more efficient (*i.e.*, requires less samples because the variances are smaller) than the original method. In this subsection we only briefly mention the most important general improvements methods. For more information, we refer to [1], [7] and [46].

- *By using antithetic variables:* apart from $X_1, \ldots, X_N$, we also generate another sample $Y_1, \ldots, Y_N$ such that $\mathrm{Cov}(X_k, Y_k) < 0$. The rationale behind this method is that $\mathrm{Var}(X_k + Y_k) = \mathrm{Var}(X_k) + \mathrm{Var}(Y_k) + 2\mathrm{Cov}(X_k, Y_k)$. Hence, if $\mathrm{Cov}(X_k, Y_k) < 0$, then we may gain in efficiency if this negativity overcompensates the effort for generating the additional Y_k. In [7] this is demonstrated for a normal density using $Y_k = -X_k$. More general, if $X_k = G(\sigma \hat{X}_k)$, where $\hat{X}_k$ are normally distributed, than

  $$X_k = G(\sigma \hat{X}_k) = G(0) + G'(0)\sigma \hat{X}_k + \mathcal{O}(\sigma^2).$$

  The mean of the linear term is zero. However, in the Monte Carlo $\sum_k X_k$ sum, the linear terms will not cancel exactly. By additionally taking $Y_k = G(-\sigma \hat{X}_k)$ into account, the linear terms do cancel. Then the remaining error really is proportional to σ^2.

- *By using control variates:* Let $\widetilde{X}$ look like X (via a coarse approximation using a limited MC-run, or from a previously obtained result, and some interpolation) and that it uses the same probability density function f as X (when using parameter-dependency this is automatically satisfied by requiring that $\widetilde{X}$ and X depend on the same parameters).
  Assume that we want to estimate $\mathcal{M} = \int x f(x) dx$ and that we know $\widetilde{\mathcal{M}} = \int \widetilde{x} f(\widetilde{x}) d\widetilde{x}$. [At first glance this may look strange. However, when dealing with functions in a parameter space, things look more natural. Then we have $\mathcal{M} = \int x(p) f(p) dp$ and $\widetilde{\mathcal{M}} = \int \widetilde{x}(p) f(p) dp$. Here $X(p)$ and $\widetilde{X}(p)$ can be different functions.]
  Then using $\sigma_Y^2 = \mathrm{Var}(Y) = \mathrm{E}[Y^2] - (\mathrm{E}[Y])^2$ for $X(p)$ and $\widetilde{X}(p)$ with p distributed

$Y = X - \lambda\widetilde{X} + \lambda\widetilde{\mathcal{M}}$ we obtain

$$\begin{aligned}
\sigma^2_{X-\lambda\widetilde{X}+\lambda\widetilde{\mathcal{M}}} &= \int [x - \lambda\widetilde{x} + \lambda\widetilde{\mathcal{M}}]^2 f(x)dx - (\mathrm{E}[Y])^2 \\
&= \int [(x-\mathcal{M}) - \lambda(\widetilde{x} - \widetilde{\mathcal{M}}) + \mathcal{M}]^2 f(x)dx - \mathcal{M}^2 \\
&= \int \{(x-\mathcal{M})^2 - 2\lambda(x-\mathcal{M})(\widetilde{x} - \widetilde{\mathcal{M}}) + \lambda^2(\widetilde{x} - \widetilde{\mathcal{M}})^2 + \\
&\quad 2[(x-\mathcal{M}) - \lambda(\widetilde{x} - \widetilde{\mathcal{M}})]\mathcal{M} + \mathcal{M}^2\} f(x)dx - \mathcal{M}^2 \\
&= \sigma^2_X - 2\lambda\mathrm{E}[(X-\mathcal{M})(\widetilde{X} - \widetilde{\mathcal{M}})] + \lambda^2\sigma^2_{\widetilde{X}} + 2\mathrm{E}[Y-\mathcal{M}]\mathcal{M} \\
&= \sigma^2_X - 2\lambda\gamma + \lambda^2\sigma^2_{\widetilde{X}}, \qquad (2.46)
\end{aligned}$$

where γ involves the correlation. Note that $\mathrm{E}[Y - \mathcal{M}] = 0$. The error in the mean of $X_k - \lambda\widetilde{X}_k$ can be much smaller than in μ. The expression (2.46) has a minimum for $\lambda = \frac{2\gamma}{2\sigma^2_{\widetilde{X}}}$, resulting in

$$\min_\lambda \sigma^2_{X-\lambda\widetilde{X}+\lambda\widetilde{\mathcal{M}}} = \sigma^2_X - \frac{\gamma^2}{\sigma^2_{\widetilde{X}}} \leq \sigma^2_X. \qquad (2.47)$$

This indicates a possible improvement. In practice one takes some chosen values for λ.

- *By a matching moment technique:* Prescribe the matching moments $m_1 = \int xF(x)dx$ and $m_2 = \int x^2 f(x)dx$ (for some, desired, density function f). Let $\mu_n(X) = \frac{1}{N}\sum_k X^n_k$, where $n > 0$, be the empirical moments (thus $\mathrm{E}[\mu_1(X)] = \mu$). We consider the transformed quantity

$$Y = \frac{X - \mu_1}{c} + m_1, \quad \text{with} \qquad (2.48)$$

$$c = \sqrt{\frac{m_2 - m_1^2}{\mu_2 - \mu_1^2}}. \qquad (2.49)$$

After also transforming X_k to Y_k we see that $\mu_n(Y) = m_n$ for $n = 1, 2$. Hence, the two first moments match. Note, however, that the Y_k are not independent (the transformation uses μ_1 and μ_2). This affects Monte Carlo error estimates (the Central Lmit Theorem is not directlty applicable) and the method may be biased.

- *By stratification:* Stratification combines randomness with the benefits of a grid.
The basic idea is to partition the space in M blocks Ω_m and to sample in each m-th block randomly distributed points X^m_k $k = 1, ..., N/M$ (with N being a multiple of M). Let $\mathcal{M}_m = |\Omega_m|^{-1}\int_{\Omega_m} xf(x)dx$ be the local mean on Ω_m. Then the error in μ (now expressed in a so-called 'stratified sum') is

$$\sigma^2_s = \sum_m \int_{\Omega_m} [x - \mathcal{M}_m]^2 f(x)dx \leq \sum_m \int_{\Omega_m} [x - \mathcal{M}]^2 f(x)dx = \sigma^2 \qquad (2.50)$$

Clearly, we need $N \gg 1$ trials, which can be rather large, but there always holds $\sigma_s \leq \sigma$. An interesting option arises if we can obtain a cheap *impression* of the distribution of the X^m_k without actually calculating them. For instance, assume that we know in some way that n_m hits fall within Ω_m, then $N = \sum_m n_m$, and the local m-th density equals

$f_m = n_m/N$. We can now actually re-sample (and evaluate) only K points $\widetilde{X}_k^m$ in each Ω_m and determine

$$\hat{\mu} \;=\; \sum_m \sum_k \widetilde{X}_k^m \frac{f_m}{K} \; . \tag{2.51}$$

If $M \times K < N$ this may become beneficial (for instance $N = 10,000$ and $M \times K = 250$). The idea can easily be applied in a parameter space that is partitioned in M blocks Ω_m. Then $X_k^m = X(p_k^m)$ is the result of X at distributed parameters p_k^m. In these cases the f_m are derived from the densities of the parameter space: indeed, they may be known in advance, or $N \gg 1$ parameters $p_{k\prime}^m$ may be generated in advance by a simulation program and can be made output before actually calculating $X_{k\prime}^m = X(p_{k\prime}^m)$ [7]. Note that also after obtaining the sampled p_k^m-s we may define the M blocks Ω_m to our convenience. We easily obtain f_m (for instance by using a histogram). After this, we re-sample only K points p_k^m in each Ω_m with $M \times K \ll N$ and evaluate $X(.)$ only at these last sample points $X_k^m = X(p_k^m)$ and apply

$$\hat{\mu} \;=\; \sum_m \sum_k X_k^m \frac{f_m}{K} \; . \tag{2.52}$$

The result is that

- less populated intervals are sampled more;
- more populated intervals are sampled less.

Stratification can easily be combined with Importance Sampling (see below) if we can obtain an impression of the distribution of the X_k^m. The procedure also offers options for refinement (hierarchically, or by Kriging, etc).

- *By importance sampling:* Note that $\mathcal{M} = \int x \frac{f(x)}{g(x)} g(x) dx$ and calculate $\hat{\mu} = \frac{1}{N} \sum_k X_k \frac{f(X_k)}{g(X_k)}$. The error in $\hat{\mu}$, using points X_k with distribution g, has

$$\sigma \;=\; \int \left[x \frac{f(x)}{g(x)} - \mathcal{M} \right]^2 g(x)\, dx \; . \tag{2.53}$$

One can thus emphasize areas where x is large. Details are described in Section 3.

In [7] techniques with control variates, or with antithetic variables are shown to reduce the error (an order). Combining the techniques result in further error reduction (half an order). Using Quasi-Monte Carlo techniques the error is slightly improved in each case, but is much faster obtained.

[7] In the circuit simulator Spectre the sampled parameters $p_{k\prime}^m$ may be generated with the 'iterVsValue' command: in fact this is a parameter distribution scan.

Section 3

Importance Sampling

When we wish to estimate extreme probabilities of the form $P(X < t)$ using the indicator random variables like in (2.32), we need a lot of samples because p is then very small (cf. (2.36). The reason is that most of our X_k's are larger than t and do not directly contribute. The main idea behind importance sampling is to circumvent this phenomenon by sampling from another distribution which has high probability to be larger than c in such a way that we may conveniently translate the results back to the originally required probability. In this section we first introduce the theory behind Importance Sampling in Subsections 3.1 and 3.2 and show explicit examples in Subsection 3.3. In the remaining subsections we describe some variants of Importance Sampling.

3.1 BACKGROUND OF IMPORTANCE SAMPLING

In this section we describe Importance Sampling, a method to increase accuracy of simulation by changing the distribution from which is sampled and suitably correcting for this change.

Suppose we are interested in probabilities of the form $p = P(Y < t)$, where Y follows a probability distribution with density function f and distribution function F. The Monte Carlo approach of Section 2.4 is based on sampling $X_1, \ldots, X_N$ from f and using the following estimator:

$$F^{\mathrm{MC}}(t) = \frac{1}{N} \sum_{i=1}^{N} I_{\{X_i < t\}} \,. \tag{3.1}$$

It follows directly from (2.33) and (2.34) that

$$\mathrm{E}_f \left(F^{\mathrm{MC}}(t) \right) = F(t) = p \text{ and } \mathrm{Var}_f \left(F^{\mathrm{MC}}(t) \right) = \frac{1}{N} F(t) \left(1 - F(t)\right) = \frac{1}{N} p(1-p). \tag{3.2}$$

For *Importance Sampling* we use an additional probability distribution with density function g, so $g(x) \geq 0$, $\int_{-\infty}^{\infty} g(x) dx = 1$. Clearly, when $g(x) > 0$ for $x \in (-\infty, t)$:

$$P(Y < t) = F(t) \;\; = \;\; \int_{-\infty}^{t} f(y)\, dy = \int_{-\infty}^{t} \frac{f(x)}{g(x)}\, g(x)\, dx. \tag{3.3}$$

On purpose we changed the dummy variables from y to x. The above formula should be read as follows: the first integral is with respect to sampling from Y which density f, while the second

integral is weighted sampling from another random variable X with density g, the weights being $f(X)/g(X)$. Importance Sampling is based on transforming the above relation into an estimator in the following way:

$$F^{\mathrm{IS}}(t) \;=\; \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{g(X_i)}\,I_{\{X_i<t\}}, \tag{3.4}$$

with the X_i chosen according to the density g, rather than to f. Note that $F^{\mathrm{IS}}(t)$ is an *unbiased* estimator for $P(Y<t)$:

$$\begin{aligned}
\mathrm{E}_g\left(F^{\mathrm{IS}}(t)\right) &= \frac{1}{N}\sum_{i=1}^{N}\mathrm{E_g}\left(I_{\{X_i<t\}}\frac{f(X_i)}{g(X_i)}\right)\\
&= \frac{1}{N}\sum_{i=1}^{N}\int_{-\infty}^{\infty} I_{\{x_i<t\}}\frac{f(x_i)}{g(x_i)}\,g(x_i)\,dx_i\\
&= \frac{1}{N}\sum_{i=1}^{N}\mathrm{E}_f(I_{\{X_i<t\}})\\
&= \frac{1}{N}\sum_{i=1}^{N}F(t)\\
&= F(t).
\end{aligned}$$

Note that this re-sampling may already be a benefit: sampling according to a known and simple g may be more efficient than sampling according to a density f that involves more calculations. However, we will now compute the variance of this estimator to study whether it is indeed more efficient than the crude Monte Carlo presented in Section 2.4. To derive an expression for $\mathrm{Var}\left(F^{\mathrm{IS}}(t)\right)$, we use (2.5) and observe that

$$\begin{aligned}
\mathrm{Var}_g\left(I_{\{X_i<t\}}\,\frac{f(X_i)}{g(X_i)}\right) &= \mathrm{E}_g\left(\left(I_{\{X_i<t\}}\frac{f(X_i)}{g(X_i)}\right)^2\right)-\mathrm{E}_g^2\left(I_{\{X_i<t\}}\frac{f(X_i)}{g(X_i)}\right)\\
&= \int_{-\infty}^{\infty}\left(I_{\{x_i<t\}}\,\frac{f(x_i)}{g(x_i)}\right)^2 g(x_i)\,dx_i - F^2(t)\\
&= \int_{-\infty}^{\infty}\left(I_{\{x<t\}}\,\frac{f(x)}{g(x)}-F(t)\right)^2 g(x)\,dx. \qquad (3.5)
\end{aligned}$$

Since X_i and X_j are independent for $i\neq j$, we also have that $I_{\{X_i<t\}}$ and $I_{\{X_j<t\}}$ are independent. Hence, because $F(t)=\int_{-\infty}^{t}f(x)dx=\int_{-\infty}^{\infty}I_{\{x<t\}}f(x)dx$,

$$\begin{aligned}
\mathrm{Var}_g\left(F^{\mathrm{IS}}(t)\right) &= \mathrm{Var}_g\left(\frac{1}{N}\sum_{i=1}^{N}I_{\{X_i<t\}}\,\frac{f(X_i)}{g(X_i)}\right)\\
&= \frac{1}{N}\left(\int_{-\infty}^{\infty}\left(I_{\{x<t\}}\,\frac{f(x)}{g(x)}-F(t)\right)^2 g(x)\,dx\right) \qquad (3.6)\\
&= \frac{1}{N}\left(\int_{-\infty}^{\infty}I_{\{x<t\}}\,\frac{f^2(x)}{g(x)}\,dx-2F(t)\int_{-\infty}^{\infty}I_{\{x<t\}}f(x)dx+F^2(t)\int_{-\infty}^{\infty}g(x)dx\right)\\
&= \frac{1}{N}\left(\int_{-\infty}^{\infty}I_{\{x<t\}}\,\frac{f^2(x)}{g(x)}\,dx-F^2(t)\right) \qquad (3.7)
\end{aligned}$$

Hence

$$\begin{aligned} N\,\mathrm{Var}_g\left(F^{\mathrm{IS}}(t)\right) &= \int_{-\infty}^{t} \frac{f^2(x)}{g(x)}\,dx - \int_{-\infty}^{t} F^2(t)g(x)\,dx - \int_{t}^{\infty} F^2(t)g(x)\,dx \\ &= \int_{-\infty}^{t} \left(\frac{f(x)}{g(x)} - F(t)\right)^2 g(x)\,dx + 2F(t)\int_{-\infty}^{t} f(x)\,dx - \\ &\quad \int_{-\infty}^{t} F^2(t)g(x)\,dx - \int_{-\infty}^{t} F^2(t)g(x)\,dx - \int_{t}^{\infty} F^2(t)g(x)\,dx \\ &= \int_{-\infty}^{t} \left(\frac{f(x)}{g(x)} - F(t)\right)^2 g(x)\,dx + 2F^2(t) - \\ &\quad 2\left(\int_{-\infty}^{t} F^2(t)g(x)\,dx + \int_{t}^{\infty} F^2(t)g(x)\,dx\right) + \int_{t}^{\infty} F^2(t)g(x)\,dx \\ &= \int_{-\infty}^{t} \left(\frac{f(x)}{g(x)} - F(t)\right)^2 g(x)\,dx + \int_{t}^{\infty} F^2(t)g(x)\,dx. \end{aligned} \tag{3.8}$$

Here (3.6)-(3.8) are three equivalent formulations. It follows from (3.8) that if one could choose $g(x) = 0$ for $x > t$ and $\frac{f(x)}{g(x)} = F(t)$ for $x < t$ (note that this choice of g indeed yields a density), then the variance of the estimator would be zero. This is not surprising, since then the estimator is constant and hence, its variance is zero. In practice we cannot implement this perfect choice, since it requires knowledge of the quantity $F(t)$ that we are trying to estimate. So preferably one should have $g(x) \approx 0$ for $x > t$, and $\frac{f(x)}{g(x)} \approx F(t)$ for $x < t$ (i.e. constant in x). In order to achieve this one usually applies an estimate for $F(t)$ and restricts oneself to $x_{\mu_g} = \mathrm{E}(g(X))$, or one minimizes the normalized standard deviation $\mathrm{Var}\left(F^{\mathrm{IS}}(t)\right)/\mathrm{E}\left(F^{\mathrm{IS}}(t)\right)$.
Note that if $\frac{f(x)}{g(x)} \leq 1$ on $(-\infty, t]$, then (3.7) implies improvement

$$\int_{-\infty}^{\infty} I_{\{x<t\}} \frac{f^2(x)}{g(x)}\,dx \leq \int_{-\infty}^{\infty} I_{\{x<t\}} f(x)\,dx = F(t) \Rightarrow \mathrm{Var}_g\left(F^{\mathrm{IS}}(t)\right) \leq \mathrm{Var}_f\left(F^{\mathrm{MC}}(t)\right). \tag{3.9}$$

For $\frac{f(x)}{g(x)} \leq \kappa \leq 1$ on $(-\infty, t]$ we find

$$\mathrm{Var}_g\left(F^{\mathrm{IS}}(t)\right) \leq \kappa\,\mathrm{Var}_f\left(F^{\mathrm{MC}}(t)\right) - \frac{1-\kappa}{N}F^2(t). \tag{3.10}$$

This means that the error estimate only slightly improves: $\sigma_g^{IS} \leq \sqrt{\kappa}\,\sigma_f^{MC}$, which for $\kappa = 0.1$ means that not an order is gained. In order to obtain more explicit comparisons in terms of required sample sizes of the crude Monte Carlo simulation of Section 2.4 with the Importance Sampling method of this section, we either have to work on a case by case basis (see Section 3.3) or to extend the Large Deviation framework of Section 2.4 to the Importance Sampling case (see Section 3.2).
For literature on Importance Sampling we refer to [8, 20, 21, 27, 29, 31, 40, 45, 48, 51].

Remark: In the general setup of importance sampling, it is assumed that the measure $\mu_g()$ induced by g is absolutely continuous with respect to the measure $\mu_f()$ induced by f (to generalize the positivity condition mentioned above):

$$\mu_f(A) = \int_A f(x)dx = 0 \implies \mu_g(A) = \int_A g(x)dx = 0. \tag{3.11}$$

A preferred additional condition is

$$\mu_f(A) \quad > \quad 0 \quad \Longrightarrow \quad \mu_g(A) \quad > \quad 0, \tag{3.12}$$

but this is not necessary. However, when one aims to also derive a cumulative probability function for several X, this assumption (3.12) becomes of interest, because it allows the re-use of the same function g.

Remark: Note that the ratio $\frac{f(x)}{g(x)}$ is in fact a Radon-Nikodym derivative of μ_g with respect to μ_f (cf. [40]).

3.2 LARGE DEVIATION BOUNDS FOR SAMPLE SIZES IN IMPORTANCE SAMPLING

In [40] Importance Sampling is applied to server systems, based on using Exponential Change of Measure (ECM). ECM is also known as Exponential Twisting, Exponential Tilting, which became popular for rare events in queueing systems. This twisting or tilting is the basic idea hidden in the Large Deviation approach that we described in Section 2.4. We refer to [6] for a detailed exposition of these ideas in the context of Importance Sampling. We define for convenience the random variables

$$V_k := I_{\{X_k<t\}} \frac{f(X_k)}{g(X_k)}. \tag{3.13}$$

Since $V_1, V_2, \ldots$ are independent and identically distributed, by the Weak Law of Large Numbers, the arithmetic mean $A_N := \frac{1}{N}\sum_{k=1}^{N} V_k$ converges to $F(t)$. However, A_N is not a Bernoulli random variable any more.
In the following we consider X having the same distribution as $X_1, X_2, \ldots$ and a corresponding $V = I_{\{X<t\}} \frac{f(X)}{g(X)}$. The moment generating function of V equals

$$\begin{aligned} \mathrm{E}_g\left[e^{\lambda V}\right] &= \int_{-\infty}^{\infty} g(x)\, e^{\lambda f(x)/g(x)}\, dx \\ &= \int_{-\infty}^{t} g(x)\, e^{\lambda f(x)/g(x)}\, dx + \int_{t}^{\infty} g(x)\, dx \\ &= \int_{-\infty}^{t} g(x)\, e^{\lambda f(x)/g(x)}\, dx + 1 - G(t), \end{aligned}$$

where $G(t) = \int_{-\infty}^{t} g(x)\, dx$. Let $\varphi(\lambda) = \ln \mathrm{E}_f\left[e^{\lambda X}\right]$. Basically we would like to proceed as in in Section 2.4. However, since we do not know the distribution of X explicitly, we have to assume something about it. For this time, we will restrict ourselves to simple *sufficient* conditions and we will not strive for full generality. Thus let X be distributed according to the distribution $\mathbb{P}$. We assume:

1. There is no $x \in \mathbb{R}$ such that $P(X = x) = 1$,
2. $\mathrm{E}_f\left[e^{\lambda X}\right] < \infty$ for all $\lambda \in \mathbb{R}$,
3. let $\mathbb{Q}_\lambda$ be the measure given by $\mathbb{P}$ with density

$$\rho_\lambda(x) = \frac{e^{\lambda x} f(x)}{\mathrm{E}_f\left[e^{\lambda X}\right]} \quad (\text{thus} \quad \int \rho_\lambda(x)\, dx = 1)$$

(which is well-defined for all $\lambda \in \mathbb{R}$ by (1)) and let Y_λ be a random variable distributed according to $\mathbb{Q}_\lambda$. We assume that

$$\mathrm{E}_{\rho_\lambda}(Y_\lambda) = \int y_\lambda \rho_\lambda(y_\lambda)\, dy_\lambda = \int y \frac{e^{\lambda y} f(y)}{\mathrm{E}_f\left[e^{\lambda Y}\right]}\, dy < \infty$$

and

$$\mathrm{Var}_{\rho_\lambda}(Y_\lambda) = \mathrm{E}\left[Y_\lambda^2\right] - \mathrm{E}^2_{\rho_\lambda}(Y_\lambda) < \infty,$$

for all $\lambda \in \mathrm{R}$.

Then, $\varphi(\lambda)$ is a two times differentiable real function with derivatives

$$\varphi'(\lambda) = \frac{\mathrm{E}_f\left[X\, e^{\lambda X}\right]}{\mathrm{E}_f\left[e^{\lambda X}\right]} = \mathrm{E}_{\rho_\lambda}(Y_\lambda), \quad \varphi''(\lambda) = \frac{\mathrm{E}_f\left[X^2\, e^{\lambda X}\right]}{\mathrm{E}_f\left[e^{\lambda X}\right]} - \frac{\mathrm{E}^2_f\left[X\, e^{\lambda X}\right]}{\mathrm{E}^2_f\left[e^{\lambda X}\right]} = \mathrm{Var}_{\rho_\lambda}(Y_\lambda).$$

If X is not supported by a single point, $\mathrm{Var}(Y_\lambda) > 0$ and φ is therefore *strictly convex*. Let

$$J(x, \lambda) = \lambda x - \varphi(\lambda). \tag{3.14}$$

As in in Section 2.4 we again consider the 'rate function'

$$I(x) = \sup_{\lambda \in \mathbb{R}} J(x, \lambda). \tag{3.15}$$

[$I(x)$ is the so-called *Legendre transform* of φ]. That implies by [11], Lemma I.4, p. 8 that $I(x)$, also is strictly (proper) convex which means that the minimizer of I is unique (if there is one). On the other hand, by the very definition of I, we have

$$I(x) \geq J(x, 0) = -\varphi(0) = -\ln\, e^0 = 0.$$

Thus, every value x with $I(x) = 0$ must be the unique minimizer of I. Now let p be as in Section 2.4. Then $I(p) = 0$, since the Strong Law of Large Numbers implies that the empirical measure of every neighbourhood of p tends to one. Hence, p is the unique minimizer of I and $I'(p) = 0$.

We assume for simplicity that the moment generating function exists for all values of $\lambda \in \mathbb{R}$. Hence, to compute the supremum in (3.15), we consider

$$\frac{d}{d\lambda} J(x, \lambda) \quad = \quad x - \frac{\mathrm{E}_g\left[V e^{\lambda V}\right]}{\mathrm{E}_g\left[e^{\lambda V}\right]}\,. \tag{3.16}$$

It seems hopeless to compute an explicit expression as in (2.43) (Bernouilli rate case) for the rate function $I(x)$ in this new generality. However, we can try to do an expansion up to second order around p. Therefore, we have to determine the values of $I(p)$, $I'(p)$ and $I''(p)$, but only $I''(p)$ is non-zero. We observe that

$$\frac{d}{d\lambda} J(x, \lambda) = 0 \quad \Longrightarrow \quad x = \Psi(\lambda), \quad \text{where} \tag{3.17}$$

$$\Psi(\lambda) \quad = \quad \frac{\int v\, e^{v\lambda} g(v)\, dv}{\int e^{v\lambda} g(v)\, dv}. \tag{3.18}$$

We note that

$$\Psi'(\lambda) \quad = \quad \frac{\int e^{v\lambda} g(v)\, dv \int v^2 e^{v\lambda} g(v)\, dv \quad - \quad [\int v e^{v\lambda} g(v)\, dv]^2}{[\int e^{\lambda} g(v)\, dv]^2}. \tag{3.19}$$

At the right-handside we can recognize an inner-product $(1, v) \equiv \int v e^{v\lambda} g(v)\, dv$. By the Cauchy-Schwarz inequality, $(1, v) \leq \sqrt{(1,1)}\sqrt{(v,v)}$ we obtain $\Psi'(\lambda) \geq 0$. This implies that Ψ is invertible and hence (3.17) defines $\lambda = \lambda(x) = \Psi^{-1}(x)$. Hence

$$I(x) \quad = \quad J(x, \lambda(x)). \tag{3.20}$$

We note that we can write

$$x \quad = \quad \Psi(\lambda) \quad = \quad \mathrm{E}_{h_\lambda}[V]\,, \tag{3.21}$$

where we define $\mathrm{E}_{h_\lambda}[V] = \mathrm{E}_g\left[Ve^{\lambda V}\right]/\mathrm{E}_g\left[e^{\lambda V}\right]$. Note that this notation as expectation is justified by defining h_λ to be the parameterized density $h_\lambda(z) = e^{\lambda v(z)} g(z)/\mathrm{E}_{\mathrm{g}}\left[e^{\lambda V}\right]$, with $v(z) = I_{\{z<t\}}\frac{f(z)}{g(z)}$. Note that $h_{\lambda=0}(z) = g(z)$.

Thus, to calculate the first (total) derivative of $I(x)$, we differentiate (3.20) with respect to x and substitute (3.17) to obtain

$$I'(x) = \lambda(x) + x\lambda'(x) - \lambda'(x)\frac{\mathrm{E}_g\left[Ve^{\lambda(x)V}\right]}{\mathrm{E}_g\left[e^{\lambda(x)V}\right]} = \lambda(x) + \lambda'(x)\left(x - \mathrm{E}_{h_\lambda}[V]\right) = \lambda(x). \tag{3.22}$$

For the second derivative of $I(x)$, we first implicitly differentiate (3.17) with respect to x which yields $1 = \frac{\partial}{\partial\lambda}\left(\mathrm{E}_{h_\lambda}(V)\right)\lambda'(x)$. The expectation in this expression can be rewritten as

$$\frac{\partial}{\partial\lambda}\left(\mathrm{E}_{h_\lambda}(V)\right) = \frac{\partial}{\partial\lambda}\frac{\mathrm{E}_g\left[Ve^{\lambda V}\right]}{\mathrm{E}_g\left[e^{\lambda V}\right]} = \frac{\mathrm{E}_g\left[V^2e^{\lambda V}\right]}{\mathrm{E}_g\left[e^{\lambda V}\right]} - \frac{\mathrm{E}_g^2\left[Ve^{\lambda V}\right]}{\mathrm{E}_g^2\left[e^{\lambda V}\right]} = \mathrm{E}_{h_\lambda}(V^2) - \mathrm{E}_{h_\lambda}^2(V) = \mathrm{Var}_{h_\lambda}(V)\,.$$

Substituting these expressions when differentiating (3.22) with respect to x, we obtain

$$I''(x) = \lambda'(x) = \frac{1}{\frac{\partial x}{\partial\lambda}} = \frac{1}{\frac{\partial}{\partial\lambda}\left(\mathrm{E}_{h_\lambda}(V)\right)} = \frac{1}{\mathrm{Var}_{h_\lambda}(V)}\,.$$

As we explained above, p is the unique minimizer of I. Since p is also an internal point, we obtain that $0 = I'(p) = \lambda(p)$. Hence,

$$I''(p) = \frac{1}{\mathrm{Var}_{h_{\lambda(p)}}(V)} = \frac{1}{\mathrm{Var}_{h_{\lambda=0}}(V)} = \frac{1}{\mathrm{Var}_g\,(V)}. \tag{3.23}$$

Similar as in Section 2.4 for deriving (2.44) we consider

$$\begin{aligned} I(p \pm \nu p) \quad &= \quad I(p) + \nu p I'(p) + \frac{1}{2}\nu^2 p^2 I''(p) + \mathcal{O}(\nu^3 p^3) \\ &= \quad \frac{1}{2}\nu^2 p^2 I''(p) + \mathcal{O}(\nu^3 p^3) \\ &= \quad \frac{\nu^2 p^2}{\mathrm{Var}_g\,(V)}. \end{aligned} \tag{3.24}$$

We obtain the following bound (3.25) for the convergence of the *importance sampling* which again stresses the important role played by the variance but which is also more accurate than the Chebyshev inequality (2.38)

$$P\left(\left|\frac{1}{N}\sum_{k=1}^{N} V_k - p\right| > \nu p\right) \leq \exp\left(-N \inf_{|x-p|>\nu p} I(x)\right) \approx \exp\left(-\frac{Np^2}{2\mathrm{Var}_g(V)}\,\nu^2\right)\,, \tag{3.25}$$

for all sufficiently large N. Indeed, if $g(x) \equiv 1$ then $V = Z$ as in (2.44) and $\mathrm{Var}_g(V) = \mathrm{Var}(Z) = \frac{1}{pq}$. Clearly we generalized (2.44).
As a corollary we can calculate the relative efficiency between crude Monte Carlo and importance sampling. Indeed, let some error probability $\alpha > 0$ be fixed. That means, we seek a bound for the number N of runs in the simulation such that

$$P\left(\left|\frac{1}{N}\sum_{k=1}^{N} V_k - p\right| > \nu p\right) \leq \alpha.$$

Using(2.44) and (3.25), we thus obtain the following conditions for the number of runs in the crude Monte Carlo (N_{MC}) and the importance sampling (N_{IS}) settings, respectively

$$\alpha = \exp[-\frac{N_{MC}p}{2q}\,\nu^2] \implies N_{MC} = \frac{2q}{p\nu^2}\,\ln(\frac{1}{\alpha}), \tag{3.26}$$

$$\alpha = \exp[-\frac{N_{IS}p^2}{2\,\mathrm{Var}_g(V)}\,\nu^2] \implies N_{IS} = \frac{2\mathrm{Var}_g(V)}{p^2\nu^2}\,\ln(\frac{1}{\alpha}). \tag{3.27}$$

This yields that the relative efficiency is given by

$$\frac{N_{IS}}{N_{MC}} = \frac{\mathrm{Var_g}(V)}{pq}.$$

Note that this is a ratio of variances, since pq is the variance for the Bernoulli variable in the crude Monte Carlo approach. Since $\mathrm{E}_g(V) = p$, we have $\mathrm{Var}_g V = \mathrm{E}_g(V^2) - \mathrm{E}_g^2(V) = \mathrm{E}_g(V^2) - p^2$. Hence, we finally obtain that the relative efficiency between the importance and the crude sampling approach is approximately given by

$$\frac{N_{IS}}{N_{MC}} = \frac{\mathrm{E}_g(V^2)}{pq} - \frac{p}{q}. \tag{3.28}$$

In (3.9) we observed that $f(x)/g(x) \leq 1$ on $(-\infty, t)$ implies a variance reduction. This yields an improved accuracy because confidence intervals for the quantity to be estimated are smaller, but it was hard to directly show efficiency in terms of the required number of runs. Because

$$\mathrm{E}_g\left(V^2\right) = \int_{-\infty}^{\infty} I_{\{x<t\}}\,\frac{f^2(x)}{g^2(x)}\,g(x)\,dx = \int_{-\infty}^{t}\frac{f(x)}{g(x)}\,f(x)\,dx \leq \int_{-\infty}^{t} f(x)\,dx = p,$$

we see that (3.28) indeed implies $N_{IS} \leq N_{MC}$ (as we expected). However we still do not have an impression on how much the actual improvement is (despite the effort in deriving (3.25)).
We can sharpen the above result a bit assuming $f(x)/g(x) \leq \kappa \leq 1$ on $(-\infty, t)$ (and $p \leq \kappa$). Then

$$\frac{N_{IS}}{N_{MC}} = \frac{\mathrm{E}_g(V^2)}{pq} - \frac{p}{q} \leq \frac{\kappa}{q} - \frac{p}{q} \leq \kappa(1+\zeta) \tag{3.29}$$

for $|(1-\frac{1}{\kappa})p + \mathcal{O}(p^2)| \leq \zeta$, which for $\kappa = 0.1$ and $p = 10^{-10}$ means that $\zeta \leq 10^{-9}$. Hence for this situation we can take an order less samples with Importance Sampling to get the same accuracy as with Monte Carlo.
The actual reduction in number of trials can only be quantified on a case-to-case base. To this end, we present explicit examples in Subsection 3.3.

Remark 4. *The gain in efficiency is merely caused by a trade-off between things that are visible such as the number of runs, and things that are invisible such as the time that your computer needs to calculate the values of* $1/\rho$ *at the sampling points. It is important to note that to achieve the wanted accuracy* νp*,* $1/\rho$ *must be effectively calculable with the same accuracy in order not to mess up the mean of the* Y_i *with the corresponding roundoff error. That also provides a limiting (though unseen) factor for the importance sampling. In practice, we will not strive for an* optimal *solution in the sense of the minimization of the variance and hence the number* N_{IS} *of runs. Because* $\mathrm{Var}_g(Y) \geq 0$ *we have* $\mathrm{E}_g[Y^2] \geq (\mathrm{E}_g[Y])^2 = p^2$ *(in fact* Jensen inequality*) and equality is assumed if and only if* Y *is almost surely constant. The corresponding optimal density in our case would be*

$$\rho_\eta^*(x) = \frac{\mathbf{1}_{f(x)>a}\rho_\xi(x)}{\int_{\{f>a\}} \rho_\xi(x')dx'} = \frac{1}{p}\rho_\xi(x)\mathbf{1}_{f(x)>a}. \tag{3.30}$$

This optimal choice of the density would be associated to the exact *solution of our problem (note that* $\mathrm{Var}\, Y^* = 0$*) and, due to* $Y^* = p$*, would even reduce* n *to one. However, we decided for a sampling approach since we realized that computing* p *directly is out of reach and hence it is infeasible to compute this optimal solution directly. Neither* p *nor the indicator* $\mathbf{1}_{f(x)>a}$ *are within reach. Hence, our approach to importance sampling should be to find densities which provide a good compromise between reducing the variance on the one hand and being effectively computable with sufficient precision on the other hand. The idea, however, will always be to use a density which is as close as possible to the ideal density.*

3.3 EXAMPLES OF IMPORTANCE SAMPLING

In this section we show some explicit examples of Importance Sampling. In particular, we show how to use normal distributions with enlarged spread and broad uniform distributions.

In sections 3.3.1-3.3.3 we basically are interested in questions like

- Is $\frac{f(x)}{g(x)} \leq 1$ in the area of interest, which is the basis for the improvement condition as formulated in (3.9)?
- Are 'natural' assumptions (3.11) and (3.12) satsified?
- What is the portion of samples in the area of interest?

In section 3.3.4 some explicit results for N_{MC} and N_{IS}, as found in literature, are summarized.

In all our examples we use $p = 10^{-10}$, $\varepsilon = 0.1p = 10^{-11}$ (so $\nu = 0.1$), and $\alpha = 0.05$. The values of N (the minimum required of simulation runs) follow directly from (3.26)-(3.27), where V has the same distribution as the variables defined by (3.13).

3.3.1 EXAMPLE: NORMAL DISTRIBUTION WITH ENLARGED SPREAD

From the above it is clear that for $f \equiv \mathrm{N}(\mu, \sigma)$ one must not expect automatically good or efficiently obtained results by taking $g \equiv \mathrm{N}(\mu, \kappa\sigma)$ with $\kappa \gg 1$ (which allows to sample also

points much further away from μ than done by f; in fact this is a form of scaling: $g(x) = \frac{1}{\kappa} f(\frac{x}{\kappa})$). For all x we have

$$\frac{f(x)}{g(x)} = \kappa\, e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2 + \frac{1}{2}(\frac{x-\mu}{\kappa\sigma})^2} = \kappa\, e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2\left[1-\frac{1}{\kappa^2}\right]} = \mathcal{O}\left(\kappa\, e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}\right) \quad (\kappa \to \infty). \tag{3.31}$$

Of course, we have $\frac{f(x)}{g(x)} \to 0$ when $(\frac{x-\mu}{\sigma})^2 \to \infty$, but $\int_{-\infty}^{t} f^2(x)/g(x)\, dx$ should be considerably smaller than $F(t)$ in order to have a significantly smaller variance (cf. (3.7)), and hence have a smaller number of required simulation runs. In other words, in general the importance sampling approach may not generate relatively many more samples in the area we are interested in than outside that area.

We consider this more closely. It appears that we are able to guarantee that $\frac{f(x)}{g(x)} \leq 1$ in the area we are interested in. We assume $x \leq t \leq t' < \mu$ and write $\frac{\mu - t'}{\sigma} = \eta\kappa$ and assume $0 \leq \eta \leq 1$ and $\kappa \geq 1$. We note that

$$\begin{aligned}
\forall x < t': \quad \frac{f(x)}{g(x)} \leq 1 \quad &\Longleftrightarrow \quad \forall x < t': \quad \kappa\, e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2\left[1-\frac{1}{\kappa^2}\right]} \leq 1 \\
&\Longleftrightarrow \quad \kappa\, e^{-\frac{1}{2}(\frac{t'-\mu}{\sigma})^2[1-\frac{1}{\kappa^2}]} \leq 1 \\
&\Longleftrightarrow \quad \kappa\, e^{-\frac{1}{2}\theta^2\kappa^2[1-\frac{1}{\kappa^2}]} \leq 1 \\
&\Longleftrightarrow \quad 2\ln(\kappa) - \theta^2(\kappa^2 - 1) \leq 0 \\
&\Longleftrightarrow \quad \theta^2 \leq \frac{2\ln(\kappa)}{(\kappa^2 - 1)} \\
&\Longrightarrow \quad 1 \geq \theta^2 \geq h(k^2), \quad \text{for} \quad h(z) = \frac{\ln(z)}{z-1}\,.
\end{aligned} \tag{3.32}$$

Clearly $\lim_{z \downarrow 1} h(z) = 1$, while $h'(z) \leq 0$ for $z \geq 1$. For $\kappa = 6.4$ (which corresponds to $P(X < t) \leq 10^{-10}$) this means $1 \geq \theta^2 \geq 0.0929$, and thus $1 \geq \theta \geq 0.3048$. Sampling with $g \equiv \mathrm{N}(\mu, 6.4\sigma)$ and taking $t' = \mu - 6\sigma$ implies that $\theta = 6/6.4 = 0.9375$, which is acceptable (note that t' corresponds with $P(X < t) \leq 10^{-9}$, see Table 2.1).
For this combination of distributions both assumptions (3.11) and (3.12) are satisfied.
We observe that a significant fraction

$$\frac{1}{\sqrt{2\pi}}\frac{1}{6.4\sigma}\int_{X=\mu-6.4\sigma}^{\infty} e^{-\frac{1}{2}\frac{(x-\mu)^2}{6.4^2\sigma^2}} dx \quad = \quad \frac{1}{\sqrt{2\pi}}\int_{-1}^{\infty} e^{\frac{x^2}{2}} dx \quad = \quad 0.8413 \tag{3.33}$$

(calculated in Matlab with 1 - normcdf(-1,0,1)) is sampled outside the area $x < t$, which may be rather disappointing (note that this will be even more when dealing with higher dimensions), but 15% falls within. Surprisingly even this already is much better than what will be needed when using ordinary Monte Carlo.

3.3.2 EXAMPLE: NORMAL DISTRIBUTION WITH SHIFTED MEAN

Another option is using shifting: $g(x) = f(x - T)$, say with $T = \kappa\sigma$ [in one of the examples in 3.3.4 below this is actually done with $\kappa = 2$]. Indeed, this is a better option, as will be shown

next. Let $f \equiv \mathrm{N}(\mu, \sigma)$ and $g \equiv \mathrm{N}(\mu - \kappa\sigma, \sigma)$. Assuming $x \leq t = \mu - \kappa\sigma < \mu$ we have

$$\begin{aligned} \forall x < t: \quad \frac{f(x)}{g(x)} \leq 1 \quad &\Longleftrightarrow \quad \forall x < t: \quad e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2} + \frac{1}{2}\frac{(x-\mu+\kappa\sigma)^2}{\sigma^2}} \leq 1 \\ &\Longleftrightarrow \quad e^{\frac{1}{2\sigma^2}[2(t-\mu)\kappa\sigma + \kappa^2\sigma^2]} \leq 1 \\ &\Longleftrightarrow \quad 2(t-\mu)\kappa\sigma + \kappa^2\sigma^2 \leq 0 \\ &\Longleftrightarrow \quad t + \frac{\kappa}{2}\sigma \leq \mu\,, \end{aligned} \tag{3.34}$$

which clearly is satisfied since $t = \mu - \kappa\sigma$.
For this combination of distributions assumption (3.11) is satisfied, but (3.12) not.
By construction, we have that a fraction

$$\frac{1}{\sqrt{2\pi}\sigma} \int_t^\infty e^{-\frac{1}{2}\frac{(x-[\mu-\kappa\sigma])^2}{\sigma^2}} dx \quad = \quad 0.5 \tag{3.35}$$

is sampled outside the area $x < t$.

3.3.3 EXAMPLE: UNIFORM DISTRIBUTION ON A BROAD INTERVAL

Next we consider the case of $f \equiv \mathrm{N}(\mu, \sigma)$ and $g \equiv \mathrm{Unif}(\mu - \kappa\sigma, \mu + \kappa\sigma)$ with $\kappa \geq 1$. Hence, $g(x) = \frac{1}{2\kappa\sigma}$ for $x \in [\mu - \kappa\sigma, \mu + \kappa\sigma]$ and $g(x) = 0$ elsewhere. Clearly, to get samples one must have $\mu - \kappa\sigma \leq x \leq t \leq t' < \mu$. As before we write $\frac{|t'-\mu|}{\sigma} = \theta\kappa$ and assume $1 \geq \theta \geq 0$ and $\kappa \geq 1$. We find

$$\begin{aligned} \forall x < t': \quad \frac{f(x)}{g(x)} \leq 1 \quad &\Longleftrightarrow \quad \forall x < t': \quad \frac{2\kappa}{\sqrt{2\pi}}\, e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \leq 1 \\ &\Longleftrightarrow \quad \frac{2\kappa}{\sqrt{2\pi}}\, e^{-\frac{1}{2}(\frac{t'-\mu}{\sigma})^2} \leq 1 \\ &\Longleftrightarrow \quad \frac{2\kappa}{\sqrt{2\pi}}\, e^{-\frac{1}{2}\theta^2\kappa^2} \leq 1 \\ &\Longleftrightarrow \quad 1 \geq \theta^2 \geq \frac{\ln(\frac{2}{\pi}) + \ln(\kappa^2)}{\kappa^2}. \end{aligned} \tag{3.36}$$

We note that $h(z) = \frac{\ln(\frac{2}{\pi}) + \ln(z)}{z}$ has a maximum $\frac{2}{e\pi} < 1$ at $z = \frac{e\pi}{2}$. Taking $\kappa = 6.4$ (as in the previous case) leads to $1 \geq \theta^2 \geq 0.0796$, and thus $1 \geq \nu \geq 0.2822$. However, this is not enough to ensure samples in the area $x < \mu - 6.4\sigma$. Hence, we extend the interval of g to $\kappa = 8$: thus $t = \mu - 8\sigma$ and $t' = \mu - 6.4\sigma$. Note that $\theta = 6.4/8 = 0.8$, which is acceptable.

Sampling with $g \equiv \mathrm{Unif}(\mu - 8\sigma, \mu + 8\sigma)$ means that a fraction

$$\int_{t'}^{\mu+8\sigma} g(x)dx \quad = \quad \int_{\mu-6.4\sigma}^{\mu+8\sigma} \frac{1}{16\sigma} dx \quad = \quad 0.9$$

is sampled outside the area $[t, t']$, which is even slightly more worse than in the case of the broad, but unshifted, normal distribution. However, again, it is much more efficient than what will be needed when using ordinary Monte Carlo.

Of course, taking a uniform distribution on a shifted interval, say $g \equiv \mathrm{Unif}(t, \sigma)$ for $t = \mu - \kappa\sigma$ will result in a more efficient method in which case only a fraction of 0.5 of the samples will be outside the area we are interested in.

3.3.4 FURTHER EXAMPLES

In [31] it is assumed that $f(x) \sim \mathrm{N}(10,2)$ and $t = 6.7 = 3.35\sigma_f$, which gives $F(t) = 0.049471 \leq 0.05$. Thus to guarantee $100 = N\,F(t)$ hits, $N_{\mathrm{MC}} = 100/F(t) \approx 2000$. When $g(x) \sim \mathrm{N}(t,2)$ one will have $I_{\{x_i<t\}} = 1$ for approximately 50% of the x_i. This time only $N_{\mathrm{IS}} = 130$ is enough to obtain a similar accuracy using importance sampling.
When $t = 3.0 = 1.5\sigma_f$ one has $F(t) = 0.00023623 \leq 0.25\ 10^{-3}$, $N_{\mathrm{MC}} \approx 4\,10^5$, and $N_{\mathrm{IS}} = 500$.
A further improvement was found by starting with a non-normal density function g

$$g(x) \quad = \quad \begin{cases} ae^{-a(t-x)} & \text{if} \quad x \leq t \\ 0 & \text{if} \quad x > t \end{cases}, \tag{3.37}$$

in which a was a free parameter that can be optimized. Clearly $\mu_g = t - \frac{1}{a}$. Equating $\frac{f(x)}{g(x)}\Big|_{x=\mu_g} = P_f(X<t) = p_f(X)$ for $t = 6.7$ gives $a \approx 1.25$. Now already for $N_{IS} = 20$ one obtains a 95% confidence interval.

In [12] a similar problem is discussed using $f(x) = e^{-x}$ (power density of thermal noise in an electronic signal) and $g_{\exp}(x) = \frac{1}{a}e^{-\frac{x}{a}}$ and $p_f(t) = \int_t^\infty f(x)dx = P(X > t)$. Here a is determined to minimize the normalized standard deviation

$$\frac{\sigma[p_f^{IS}(t)]}{\mathrm{E}[p_f^{IS}(t)]} \quad = \quad \frac{1}{\sqrt{N_{IS}}}\sqrt{\frac{a^2}{2a-1}e^{t/a} - 1}\,, \tag{3.38}$$

which gives as optimal values

$$a_{1,2} \quad = \quad \frac{1}{2}[1 + t \pm \sqrt{1+t^2}] \tag{3.39}$$

(below we will take the "+" sign in the calculations). For ordinary Monte Carlo the normalized standard deviation is

$$\frac{\sigma[p_f^{MC}(t)]}{\mathrm{E}[p_f^{MC}(t)]} \quad = \quad \frac{1}{\sqrt{N}}\sqrt{e^t - 1}\,. \tag{3.40}$$

By equating (3.38) and (3.40) one can consider the simulation gain by Importance Sampling when compared to Monte Carlo to obtain the same minimum normalized standard deviation

$$\frac{N_{MC}}{N_{IS}} \quad = \quad \frac{e^t - 1}{\frac{a^2}{2a-1}e^{t/a} - 1} \quad > \quad 1. \tag{3.41}$$

For $t = 8$ and $N_{MC} = 10^4$ we find $\frac{N_{MC}}{N_{IS}} = 282$, i.e $N_{IS} = 35$.
For $t \gg 1$ we have that $a \approx t$ and thus

$$\frac{N_{MC}}{N_{IS}} \quad \approx \quad \frac{2e^t}{e\,t} \quad \gg \quad 1. \tag{3.42}$$

For $t \approx 20$, one has (using $e^3 \approx 20$ and $2^{10} \approx 10^3$) that $\frac{N_{MC}}{N_{IS}} \approx 10^8$, which means an enormous speed up.

In [12] also other examples are given

- $f(x) = \frac{1}{K!}x^K e^{-x}$ [gamma-distribution for a sum of K exponential samples, with shape parameter K] for $p_f(t) = \int_t^\infty f(x)dx = P(X > t)$ with $2 \leq 10\log_{10}(t) \leq 12$ and $K = 2, 5, 10, 20$ by Importance Sampling using $g(x)$ as in (3.37) with $a = t/\ln(2)$ (giving $g(x) = a\,2^{-a^2}\,e^{ax}$, and $g_{\rm box}$, defined by

$$g_{\rm box}(x) \;=\; \begin{cases} \frac{1}{t} & \text{if} \quad t \leq x \leq 2t \\ 0 & \text{else} \end{cases} \tag{3.43}$$

(the latter using samples $x_k = t(1 + u_k)$, with $u_k \in [0, 1]$ uniformly). In this case the $g_{\rm box}$-function is the most efficient one, but also offers the opportunity to take simple uniform variates as samples: 26% of the $g_{\rm box}$-sampling are in the area that contributes 99% to $p_f(t)$; for $g_{\rm exp}(x)$ this is 8%. [No speed-up when compared to standard MC is mentioned]

- $f(x) = \frac{4}{\sqrt{\pi}}x^2 e^{-x^2}$ [Maxwellian distribution of molecular speeds in a dilute gas]. Using

$$g_{\rm box}(x) \;=\; \begin{cases} \frac{1}{2} & \text{if} \quad t \leq x \leq t+2 \\ 0 & \text{else} \end{cases} \quad , \tag{3.44}$$

results for $t = 4$ and for $t = 6$ are obtained. An additional remark is that points $z_{k'}$ can be generated that are "Maxwellian distributed": If $I_k \equiv I(x > t) > [f(t)/g_{\rm box}(t)]u_k$ then "$k' = k' + 1$ and $z_{k'} = x_k$". This can be an additional benefit.

3.4 MULTIVARIATE IMPORTANCE SAMPLING

In this section we briefly describe how to use importance sampling in a multivariate setting.

In several simulations the nonlinear output response $x(p)$ depends on independent input parameters with known density distribution function (in most cases a normal distribution). In this case the ratio $f(p)/g(p)$ is considered in p-space, where f is known and thus the ratio can easily be calculated. Of course, in a more dimensional parameter space the definition of $g(p)$ that should cover the area of parameters for the rare events of interest, requires more attention. Multivariation also naturally introduces effects due to dimensionality as is seen in the examples below. With increasing dimension of the parameter space importance sampling becomes more important.

In [31] $x(p_1, p_2) = \sqrt{p_1^2 + 3p_2^2}$ was considered and samples $x_i = x((p_1, p_2)_i) = x(p_{1i}, p_{2i})$ in which the input parameters p_{ki} are chosen according to density f_k.
Now $p_f(X) = \iint\limits_{X_p} f_1(p_1)f_2(p_2)dp_1dp_2$, in which X now is identified with a 2-D area X_p in (p_1, p_2) such that $x(p_1, p_2) > X$ (or $< X$).
The indicator function is now defined by

$$I_X(x) = I_{X_p}(p_1, p_2)) = \begin{cases} 1 & \text{if} \quad (p_1, p_2) \in X_p, \;\; \text{i.e. if} \;\; x(p_1, p_2) > X \\ 0 & \text{else} \end{cases} \tag{3.45}$$

and similar as in (2.28) one can estimate $p_f(X)$ by

$$p_f^{\rm MC}(X) \;\approx\; \frac{1}{N}\sum_{i=1}^{N} I_X(x_i). \tag{3.46}$$

The counter parts of (3.6)-(3.8) are

$$\begin{aligned}
\mathrm{Var}_g[p_f^{\mathrm{IS}}(X)] &= \frac{1}{N}[\iint \{I_{X_p}(p_1,p_2)\frac{f_1(p_1)f_2(p_2)}{g_1(p_1)g_2(p_2)} - p_f(X)\}^2 g_1(p_1)g_2(p_2)dp_1dp_2] && (3.47)\\
&= \frac{1}{N}[\iint_{X_p} [\frac{f_1(p_1)f_2(p_2)}{g_1(p_1)g_2(p_2)}]^2 g_1(p_1)g_2(p_2)dp_1dp_2 - p_f^2(X)] && (3.48)\\
&= \frac{1}{N}[\iint_{X_p} [\frac{f_1(p_1)f_2(p_2)}{g_1(p_1)g_2(p_2)} - p_f(X)]^2 g_1(p_1)g_2(p_2)dp_1dp_2 + \\
&\quad \iint_{\mathbb{R}^2\setminus X_p} p_f^2(X) g_1(p_1)g_2(p_2)dp_1dp_2]. && (3.49)
\end{aligned}$$

In [31] $f_1 \sim \mathrm{N}(\mu_{f_1} = 20, \sigma_{f_1} = 2)$, and $f_1 \sim \mathrm{N}(\mu_{f_2} = 10, \sigma_{f_2} = 1)$ was taken. Note that for $p_1 = \mu_{f_1} + 2\sigma_{f_1} = 24$ and $p_2 = \mu_{f_2} + 2\sigma_{f_2} = 12$, one has $x(p_1, p_2) = 12\sqrt{7} \approx 31.75$. For $X = 32$, $p_f(X) = P(x(p_1, p_2) \geq 32) \approx 0.18\ 10^{-2}$.
For importance sampling two functions g_1, g_2 can be defined: $g_1 \sim \mathrm{N}(\mu_{g_1} = 24, \sigma_{g_1} = 2)$, $g_2 \sim \mathrm{N}(\mu_{g_2} = 12, \sigma_{g_2} = 1)$. For $p_f^{\mathrm{IS}}(X)$, the 95% confidence interval at $N_{IS} = 20$ was already comparable to the one for $p_f^{\mathrm{MC}}(X)$ at $N_{MC} = 2000$.
Trying $g_1 \sim \mathrm{N}(\mu_{g_1} = \mu, \sigma_{g_1} = 2)$, $g_2 \sim \mathrm{N}(\mu_{g_2} = \mu/2, \sigma_{g_2} = 1)$ an optimum value $\mu = 25$ was found, but this did not much improve the results furthermore and neither did improve the efficiency.

3.5 WEIGHTED IMPORTANCE SAMPLING

Hesterberg [20] describes two additional variants on Importance Sampling: the "Ratio" or "Weighted" Importance Sampling method and the Regression Importance Sampling method.
The "Ratio" or "Weighted" Importance Sampling is defined by

$$F^{\mathrm{WIS}}(t) = \frac{\frac{1}{N}\sum_{i=1}^N V_i}{\frac{1}{N}\sum_{i=1}^N W_i} = \frac{\overline{V}}{\overline{W}}, \qquad (3.50)$$

where $V(X) = I_{X<t}(X)W(X)$ and $W(X) = f(X)/g(X)$, and with this $V_i = V(X_i)$ and $W_i = W(X_i) = f(X_i)/g(X_i)$. Clearly, $W(X)$ has expectation $\mathrm{E}_g[W(X)] = 1$. In particularly we have $\mathrm{E}_g[W_i] = \mathrm{E}_g[W(X_i)] = 1$.
If in $V(X)$ the function $I_{X<t}(X)$ is written as a sum $I_{X<t}(X) = A(X) + c$ (assuming fixed t and a constant c), the Weighted Importance Sampling result for the sum is the corresponding one for A shifted by c. For the normal Importance Sampling this only holds for the expectations. The price to be paid, however, is a (small) biasing of the expectation. Each V_i has $p = \mathrm{E}_g[V_i]$. Let $\widetilde{V}(X) = V(X) - p$, then $\widetilde{V}_i = V(X_i) - p = V_i - p$, and similarly $\widetilde{V}(X) = V(X) - p$, and similarly $\widetilde{W}(X) = W(X) - 1$, implying $\widetilde{W}_i = W(X_i) - 1 = W_i - 1$. Then

$$\begin{aligned}
F^{\mathrm{WIS}}(t) &= \frac{\frac{1}{N}\sum_{i=1}^N \widetilde{V}_i + p}{\frac{1}{N}\sum_{i=1}^N \widetilde{W}_i + 1} = \frac{\overline{\widetilde{V}} + p}{\overline{\widetilde{W}} + 1}\\
&= p\left(1 + \frac{\overline{\widetilde{V}}}{p}\right)\left(1 - \overline{\widetilde{W}} + \left[\overline{\widetilde{W}}\right]^2 + \ldots\right)\\
&= p + \left[\overline{\widetilde{V}} - p\overline{\widetilde{W}}\right] - \left[\overline{\widetilde{V}}\,\overline{\widetilde{W}} - p\left[\overline{\widetilde{W}}\right]^2\right] + \ldots \qquad (3.51)
\end{aligned}$$

The second term has g-expectation 0, but its variance is not. Using the independency of the X_i and a χ^2-related argument for the remaining products we derive the following expressions for the dominant terms in the Expectation (note that $\mathrm{E}_g[[\overline{\widetilde{W}}\,]^2] = \frac{1}{N}\mathrm{E}_g[\widetilde{W}^2]$, etc)

$$\begin{aligned}
\mathrm{E}_g[F^{\mathrm{WIS}}(t)] &\approx p - \mathrm{E}_g[\overline{\widetilde{V}}\;\overline{\widetilde{W}} - p[\,\overline{\widetilde{W}}\,]^2] \\
&= p - \frac{1}{N}\mathrm{E}_g[\widetilde{V}(X)\widetilde{W}(X) - p\widetilde{W}^2(X)] \\
&= p - \frac{1}{N}\mathrm{E}_g[\{\widetilde{V}(X) - p\widetilde{W}(X)\}\widetilde{W}(X)] \\
&= p - \frac{1}{N}\mathrm{E}_g[\{(V(X)-p) - p(W(X)-1)\}(W(X)-1)] \\
&= p - \frac{1}{N}\mathrm{E}_g[\{V(X) - pW(X)\}W(X)], \qquad (3.52)
\end{aligned}$$

where we used that $\mathrm{E}_g(V) = p$. Similarly, for the dominant terms in the Variance we derive

$$\begin{aligned}
\mathrm{Var}_g[F^{\mathrm{WIS}}(t)] &= \mathrm{Var}_g[\,\overline{\widetilde{V}} - p\overline{\widetilde{W}}\,] \\
&= \frac{1}{N}\mathrm{Var}_g[\widetilde{V} - p\widetilde{W}] \\
&= \frac{1}{N}\mathrm{Var}_g[V(X) - p - p(W(X)-1)] \\
&= \frac{1}{N}\mathrm{Var}_g[V(X) - pW(X)]. \qquad (3.53)
\end{aligned}$$

Note that in (3.52) and (3.53) $V(X) = I_{X<t}(X)W(X)$, which seems tentative, because we can split of a factor $W(X)$. However, the expectation is w.r.t. g, rather than to f.

In [26] the Weighted Importance Sampling method recently has been applied to SRAM yield simulations.

3.6 REGRESSION IMPORTANCE SAMPLING

Hesterberg [20] also describes the Regression Importance Sampling method. Similar as in the previous section we have $V(X) = I_{X<t}(X)W(X)$ and $W(X) = f(X)/g(X)$, and with this $V_i = V(X_i)$ and $W_i = W(X_i) = f(X_i)/g(X_i)$. Again, $\mathrm{E}_g[W(X)] = 1$ and $\mathrm{E}_g[W_i] = \mathrm{E}_g[W(X_i)] = 1$.
Let $\mathbf{v} = (V_1, \ldots, V_N)^T$, $\mathbf{w} = (W_1, \ldots, W_N)^T$. Hesterberg [20] considers regression on (V_i, W_i) to obtain $Z = \gamma^\star W + \delta^\star$, in which $\gamma^\star$, $\delta^\star$ are determined by regression. Because for each W_i one has $\mathrm{E}_g[W_i] = 1$, as optimum value as estimator by the Regression Importance Sampling Method the value

$$F^{MCRIS}(t) \;:=\; Z(W=1) \;=\; \gamma^\star + \delta^\star \qquad (3.54)$$

is taken. We describe the method in some more details.
Let $\mathbf{1} = (1, \ldots, 1)^T$, $\mathbf{A} = (\mathbf{w}\;\; \mathbf{1})$, $\mathbf{x} = (\gamma\;\; \delta)^T$ and $\mathbf{x}^\star = (\gamma^\star\;\; \delta^\star)^T$.
We determine $\mathbf{x}^\star$ such that $||\mathbf{v} - \mathbf{A}\mathbf{x}||^2$ is minimum. Note that $(\,\mathbf{A}^T\mathbf{v}\,)^T = (\mathbf{w}^T\mathbf{v}\,, N\overline{V}\,)^T$.

Clearly

$$\begin{aligned}
\mathbf{x}^\star &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{v} \\
&= \frac{1}{\mathbf{w}^T\mathbf{w}N - N^2[\,\overline{W}\,]^2}\begin{pmatrix} N & -N\overline{W} \\ -N\overline{W} & \mathbf{w}^T\mathbf{w} \end{pmatrix}\begin{pmatrix} \mathbf{w}^T\mathbf{v} \\ N\overline{V} \end{pmatrix} \\
&= \frac{1}{\mathbf{w}^T\mathbf{w}N - N^2[\,\overline{W}\,]^2}\begin{pmatrix} N\mathbf{w}^T\mathbf{v} - N^2\,\overline{W}\,\overline{V} \\ -N\mathbf{w}^T\mathbf{v}\overline{W} + N\mathbf{w}^T\mathbf{w}\overline{V} \end{pmatrix} \\
&= \frac{1}{\frac{1}{N}\mathbf{w}^T\mathbf{w}N - [\,\overline{W}\,]^2}\begin{pmatrix} \frac{1}{N}\mathbf{w}^T\mathbf{v} - \overline{W}\,\overline{V} \\ -\frac{1}{N}\mathbf{w}^T\mathbf{v}\overline{W} + \frac{1}{N}\mathbf{w}^T\mathbf{w}\overline{V} \end{pmatrix} \\
&= \frac{1}{\hat{\sigma}_W^2}\begin{pmatrix} \mathrm{cov}(\mathbf{w},\mathbf{v}) \\ -\frac{1}{N}\mathbf{w}^T\mathbf{v}\overline{W} + (\overline{W})^2\overline{V} + [\frac{1}{N}\mathbf{w}^T\mathbf{w} - (\overline{W})^2]\overline{V} \end{pmatrix} \\
&= \frac{1}{\hat{\sigma}_W^2}\begin{pmatrix} \mathrm{cov}(\mathbf{w},\mathbf{v}) \\ \hat{\sigma}_W^2\overline{V} + (\overline{W}\,\overline{V} - \frac{1}{N}\mathbf{w}^T\mathbf{v})\overline{W} \end{pmatrix} \\
&= \frac{1}{\hat{\sigma}_W^2}\begin{pmatrix} \mathrm{cov}(\mathbf{w},\mathbf{v}) \\ \hat{\sigma}_W^2\overline{V} - \mathrm{cov}(\mathbf{w},\mathbf{v})\overline{W} \end{pmatrix} \quad = \quad \begin{pmatrix} \gamma^\star \\ \delta^\star \end{pmatrix}, \qquad (3.55)
\end{aligned}$$

where

$$\hat{\sigma}_W^2 = \frac{1}{N}\mathbf{w}^T\mathbf{w} - [\,\overline{W}\,]^2 = \overline{W^2} - (\overline{W})^2 = \frac{1}{N}\sum_{i=1}^{N}(W_i - \overline{W})^2, \qquad (3.56)$$

$$\mathrm{cov}(\mathbf{w},\mathbf{v}) = \frac{1}{N}\mathbf{w}^T\mathbf{v} - \overline{W}\,\overline{V} = \frac{1}{N}\sum_{i=1}^{N}(W_i - \overline{W})(V_i - \overline{V}). \qquad (3.57)$$

The Regression Importance Sampling method takes as estimator for $p_f(X)$ the optimum value for Z at $W = 1$

$$\begin{aligned}
F^{\mathrm{MCRIS}}(t) &:= \gamma^\star + \delta^\star \\
&= \overline{V} - \frac{\mathrm{cov}(\mathbf{w},\mathbf{v})}{\hat{\sigma}_W^2}(\overline{W} - 1) \qquad (3.58) \\
&= \overline{V} - \beta(\overline{W} - 1) = \overline{V} + \alpha\,\mathrm{cov}(\mathbf{w},\mathbf{v}) \qquad (3.59) \\
&= \frac{1}{N}\sum_{i=1}^{N}[1 + \alpha(W_i - \overline{W})]V_i, \qquad (3.60)
\end{aligned}$$

where

$$\alpha = \frac{1 - \overline{W}}{\overline{W^2} - (\overline{W})^2} = \frac{1 - \overline{W}}{\hat{\sigma}_W^2} \qquad (3.61)$$

$$\beta = \frac{\mathrm{cov}(\mathbf{w},\mathbf{v})}{\hat{\sigma}_W^2}. \qquad (3.62)$$

In [20] the following dominant terms in the Expectation and the Variance are derived

$$\mathrm{E}_g[F^{\mathrm{MCRIS}}(t)] = p - \frac{1}{N\hat{\sigma}_W^2}\mathrm{E}_g[(W-1)^2\{(V-p) - \beta(W-1)\}], \qquad (3.63)$$

$$\mathrm{Var}_g[F^{\mathrm{MCRIS}}(t)] = \frac{\mathrm{Var}_g(V - \beta W)}{N}. \qquad (3.64)$$

3.7 PARAMETERIZED IMPORTANCE SAMPLING

In [45, 51] a parameter θ is introduced in the distribution g: $g(x,\theta)$. We may choose θ such that the variance $\mathrm{Var}_g(F^{\mathrm{IS}}(t))$ is minimized. According to (3.7) this is equivalent to minimizing

$$I(\theta) \;=\; \int_{-\infty}^{\infty} I_{\{x<t\}} \frac{f^2(x)}{g(x,\theta)} dx \;=\; \mathrm{E}_f[I_{\{x<t\}} w(x,\theta)] \;=\; \mathrm{E}_g[I_{\{x<t\}} w^2(x,\theta)], \quad (3.65)$$

in which $w(x,\theta) = f(x,\theta)/g(x,\theta)$. More generally we have

$$I^{(k)}(\theta) \;=\; \mathrm{E}_f[I_{\{x<t\}} \frac{\partial^k w(x,\theta)}{\partial \theta^k}] \;=\; \mathrm{E}_g[I_{\{x<t\}} \frac{\partial^k w(x,\theta)}{\partial \theta^k}\; w(x,\theta)]. \quad (3.66)$$

(where $k = 0, 1, 2, ...$). For a given θ and a given sampling X_i according to $g(x,\theta)$ we may estimate

$$I^{(k)}(\theta) \;\approx\; \frac{1}{N} \sum_{i=1}^{N} I_{\{X_i<t\}} \frac{\partial^k w(X_i,\theta)}{\partial \theta^k}\; w(X_i,\theta)\;. \quad (3.67)$$

To minimize $I(\theta)$ we determine a stationary point $\theta^\star$ such that $I^{(1)}(\theta^\star) = 0$ by applying a Newton process to $I^{(1)}(\theta)$, which results in a sequence of points θ_n determined by the recursion

$$\theta^{(n+1)} \;=\; \theta^{(n)} - \lambda_n\; I^{(1)}(\theta^{(n)})/I^{(2)}(\theta^{(n)}). \quad (3.68)$$

Here $\lambda_n \in [0,1]$ is a damping parameter. If (3.67) is used we may speak of a 'stochastic' Newton process. Note that in this case the sampling points may have to be adapted within each Newton iteration.
In [51] an example is given using a scaled random variable, i.e. by using $g(x) = \frac{1}{a} f(\frac{x}{a})$, which gives $I(a) = \int_{-\infty}^{t} \frac{af^2(x)}{f(x/a)} dx$. Note that $I'(1) = -tf(t) < 0$, while $I(a) \to +\infty \;\; (a \to \infty)$, which implies that $I(a)$ has a minimum for $1 < a < \infty$. However, when $xf(x) \to 0$ if $x \to \infty$ we also have $F^{\mathrm{IS}}(t) \to 0$ if $a \to \infty$.
A similar remark can be made when one applies shifting or a translation, for instance by using $g(x) = f(x+c)$. Then the variance is given by $I(c) = \int_{-\infty}^{t} \frac{f^2(x)}{f(x+c)} dx$, which also has a minimum for $0 < c < \infty$.

EXAMPLE: SHIFTED AND SCALED NORMAL DISTRIBUTION
This idea we can apply to the examples in Section 3.3. Let $f \equiv \mathrm{N}(\mu,\sigma)$ and $g(\theta_1,\theta_2) \equiv \mathrm{N}(\mu - \theta_1\sigma, \theta_2\sigma)$. Hence $\theta = (\theta_1,\theta_2)^T = (T/\sigma, \kappa)^T$ represent a T/σ-shift and a κ-standard deviation on the σ-scale of the f-distribution.
We define

$$\begin{aligned} F^T(\theta) \;\equiv\; \nabla I \;&=\; (\frac{\partial I}{\partial \theta_1},\; \frac{\partial I}{\partial \theta_2}) \\ &=\; (\;\mathrm{E}_f[I_{\{x<t\}} \frac{\partial w(x,\theta_1,\theta_2)}{\partial \theta_1}],\quad \mathrm{E}_f[I_{\{x<t\}} \frac{\partial w(x,\theta_1,\theta_2)}{\partial \theta_2}]\;). \end{aligned} \quad (3.69)$$

The Newton process for finding a root of $F(\theta) = 0$ is defined by

$$Y(\theta^{(n)})\Delta^{(n+1)} \;=\; -F(\theta^{(n)}) \quad \text{in which } \; Y(\theta) = \frac{\partial F}{\partial \theta}(\theta), \quad (3.70)$$

$$\theta^{(n+1)} \;=\; \theta^{(n)} + \lambda_n\; \Delta^{(n+1)}, \quad (3.71)$$

in which $\lambda_n \in [0,1]$ is a damping parameter. Let

$$
\begin{aligned}
H_1 &= H(x,\mu,\sigma) = \frac{x-\mu}{\sigma},\\
H_2 &= H(x,\mu-\theta_1\sigma,\theta_2\sigma) = \frac{x-\mu+\theta_1\sigma}{\theta_2\sigma},\\
E_1 &= E_1(x,\mu,\sigma) := \exp[-\frac{1}{2}\ H^2(x,\mu,\sigma)] = \exp[-\frac{1}{2}H_1^2],\\
E_2 &= E_2(x,\mu,\sigma,\theta_1,\theta_2) := \exp[\frac{1}{2}\ H^2(x,\mu-\theta_1\sigma,\theta_2\sigma)] = \exp[\frac{1}{2}H_2^2],
\end{aligned}
$$

then

$$
\begin{aligned}
\frac{\partial H_2}{\partial\theta_1} &= \frac{1}{\theta_2}, \quad \frac{\partial H_2}{\partial\theta_2} = -\frac{x-\mu+\theta_1\sigma}{\theta_2^2\sigma} = -\frac{H_2}{\theta_2},\\
\frac{\partial E_2}{\partial\theta_1} &= \frac{E_2\ H_2}{\theta_2}, \quad \frac{\partial E_2}{\partial\theta_2} = -\ \frac{E_2\ H_2^2}{\theta_2}.
\end{aligned}
$$

We consider μ, σ as fixed. For $w(x,\theta_1,\theta_2) := \theta_2\ E_1(x,\mu,\sigma)\ E_2(x,\mu,\sigma,\theta_1,\theta_2)$ we derive

$$
\frac{\partial w}{\partial\theta_1} = \theta_2\ E_1\ \frac{\partial E_2}{\partial\theta_1} = E_1\ E_2\ H_2, \tag{3.72}
$$

$$
\frac{\partial w}{\partial\theta_2} = E_1\ E_2 + \theta_2\ E_1\frac{\partial E_2}{\partial\theta_2} = E_1\ E_2\ [1-H_2^2] \tag{3.73}
$$

and

$$
\frac{\partial^2 w}{\partial\theta_1^2} = E_1\ E_2\ [1+H_2^2]\ \frac{1}{\theta_2}, \tag{3.74}
$$

$$
\frac{\partial^2 w}{\partial\theta_1\partial\theta_2} = -\ E_1\ E_2\ H_2\ [1+H_2^2]\ \frac{1}{\theta_2}, \tag{3.75}
$$

$$
\frac{\partial^2 w}{\partial\theta_2^2} = E_1\ E_2\ H_2^2\ [1+H_2^2]\ \frac{1}{\theta_2}. \tag{3.76}
$$

The Hessian matrix of w equals

$$
E_1\ E_2\ [1+H_2^2]\ \frac{1}{\theta_2}\ \begin{bmatrix} 1 & -H_2 \\ -H_2 & H_2^2 \end{bmatrix}, \tag{3.77}
$$

of which the last matrix has non-negative real eigenvalues $\lambda_1 = 0$ at eigenvector $(\theta_1,\theta_2)^T = (H_2,1)^T$ and $\lambda_2 = 1+H_2^2$ at $(\theta_1,\theta_2)^T = (1,-H_2)^T$, respectively. This implies that, for $\theta_2 > 0$, the Hessian matrix of w is non-negative definite (despite a Gershgorin circle [44, p.184] around 0 for small vales of H_2) and thus the Hessian matrix of $I(\theta)$. We note that

$$
F(\theta) = \begin{bmatrix} \mathrm{E}_f[I_{\{x<t\}}E_1E_2(\theta)H_2(\theta)] \\ \mathrm{E}_f[I_{\{x<t\}}E_1E_2(\theta)(1-H_2^2(\theta))] \end{bmatrix} = \begin{bmatrix} \int_{-\infty}^t E_1^2E_2(\theta)H_2(\theta)dx \\ \int_{-\infty}^t E_1^2E_2(\theta)(1-H_2^2(\theta))dx \end{bmatrix} \tag{3.78}
$$

$$
Y(\theta) = \frac{1}{\theta_2}\begin{bmatrix} \int_{-\infty}^t E_1^2\ E_2(\theta)\ [1+H_2^2(\theta)]dx & -\int_{-\infty}^t E_1^2\ E_2(\theta)\ H_2(\theta)\ [1+H_2^2(\theta)]dx \\ -\int_{-\infty}^t E_1^2\ E_2(\theta)\ H_2(\theta)\ [1+H_2^2(\theta)]dx & \int_{-\infty}^t E_1^2\ E_2(\theta)\ H_2^2(\theta)\ [1+H_2^2(\theta)]dx \end{bmatrix} \tag{3.79}
$$

in which we dropped all parameters other than $\theta = (\theta_1,\theta_2)^T$. Note that E_1, E_2 and H_2 depend on the integration variable x.

The optimum point $\theta^\star$ depends on t and thus one may look[1] to $\frac{d}{dt}\theta^\star(t)$ when increasing t and adaptively upgrade $\theta^\star$.

The integrals in (3.78)-(3.79) are not treated well by Mathematica for $t \ll 0$. Hence additional numerical procedures are needed to calculate them accurately.

[1]Note that: $\frac{d}{dx}\int_a^x f(x,u)du = \int_a^x f_x(x,u)du + f(x,x)$.

Section 4

Adaptive Important Sampling for Tail Probabilities of Costly Functions

4.1 ADAPTIVE IMPORTANCE SAMPLING

The efficiency of Importance Sampling depends on how the distribution $g(x)$ can be chosen, or can be constructed. In [27] a non-parametric adaptive importance sampling (NAIS) procedure is described, that needs the known distribution function f. The distribution g is improved in Algorithm 1. A refinement can be to re-use also the old results x_i to get better estimates of $p_f(X)$

Algorithm 1 NAIS:Non-parametric Adaptive Importance Sampling [27]

Step 1: Let $K(x) = \begin{cases} \frac{1}{2} & \text{if} \quad ||x|| < 1 \\ 0 & \text{otherwise} \end{cases}$ be a rectangular kernel function.
Step 2: Let $h > 0$ be a smoothing parameter.
Step 3: Initialize a simulation run to collect rare event samples y_i ($i = 1, \ldots, k$). Let $\mathcal{Y} = \{y_1, \ldots, y_k\}$.
Step 4: Define an estimate of the optimal sampling distribution $g(x)$ by the "kernel function estimation method" $g(x) = \frac{1}{\#\mathcal{Y}} \sum_{y_i \in \mathcal{Y}} \frac{1}{h} K(\frac{x - y_i}{h})$.
Step 5: Generate events using the distribution $g(x)$. Apply Importance Sampling (3.4) with this. Save the rare events y_i' ($i = 1, \ldots, k'$).
Step 6: Let $\mathcal{Y} = \mathcal{Y} \cup \{y_1', \ldots, y_{k'}'\}$.
Step 7: Go to Step 4.

using the updated expression of g.
In the following subsections we develop an alternative method for finding an optimum g. The method naturally locates "bumps", maintains a distance between sample points and is adaptive in the sence that when the levels are increased the distribution function g is easily adapted. This is especially appreciated when constructing a cumulative distribution function. The method is intelligent in that it learns from internal Monte Carlo-like evaluations and adaptively adapts g.
The method can be enhanced to also deal with more functions and to offer a user-defined threshold for finding an optimum distribution g.

4.2 STATEMENT OF THE PROBLEM

Let X be an $\mathbb{R}^d$-valued random variable with probability distribution P. From now on, we will assume that X is a multivariate Gaussian variable with covariance matrix Σ and density

$$\rho(x) = \frac{1}{\sqrt{2\pi \det(\Sigma)}^d} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x\right\}$$

even though this is not important for most of the considerations below. In the sequel, we will consider the problem of exploring the distribution of tail probabilities $p_\alpha = P(h(X) > \alpha)$ for a given function $h : \mathbb{R}^d \to \mathbb{R}$ (exploring distribution of tail probabilities $p_\alpha = P(h(X) < \alpha)$ is done similarly). More precisely, we want to explore that part of the distribution for very large values of $\alpha > \alpha_0$, meaning that the probabilities p_α are small and thus difficult to estimate using simulations based on naive Monte-Carlo methods. Therefore, on the one hand we have to estimate p_α for different values of α, and on the other hand, we are forced to use importance sampling algorithms.

However, there is one additional feature which gives the problem a somewhat different flavor. Roughly speaking, we introduce the term *costly function* for a function $h : \mathbb{R}^d \to \mathbb{R}$ for which it takes quite some effort to compute the values $f(x)$ for a given $x \in \mathbb{R}^d$. We will leave this notion vague, not attempting to make it precise for instance in terms of runtime. Even more, we think of this term in a relative way in the sense that the calculation of the value of $h(x)$ is the limiting factor in every Monte-Carlo simulation which aims at estimating the tail probabilities above. Thus, the most important reason for keeping the number of simulation runs small is that it takes so much time to decide for a given $x \in \mathbb{R}^d$ whether $h(x) > \alpha$ actually holds.

In the sequel, we will think of the function h as being unknown meaning that we have no prior information about the location of the super-level sets $\mathcal{S}_\alpha := \{x \,|\, h(x) > \alpha\}$ and therefore in the beginning no indication how to choose the importance sampling distribution. That means, the importance sampling has to be *adaptive* in the sense that finding a sampling distribution is part of the algorithm.

From the remark at the end of Section 3.2, it is rather clear that we can not expect any miracles of an importance sampling algorithm. The function is difficult to compute and this will remain so. Thus, in one way or another we will have to *explore* the function and this will inevitably be costly. The reason why we believe that we can gain something – and also the basic idea underlying the algorithm we propose – is the simple observation that the super - level sets decrease monotonously, i.e. $\alpha' > \alpha$ implies $\mathcal{S}_{\alpha'} \subset \mathcal{S}_\alpha$ and that we may therefore base the exploration of the super - level set $\mathcal{S}_{\alpha'}$ on the prior knowledge already obtained by the exploration of $\mathcal{S}_\alpha$. We will make that precise in the sequel.

Remark 5. *The assumption that no prior knowledge about h is available is a worst case scenario. Every additional piece of information about h may lead to variants of the algorithm with improved efficiency. However, we believe that the basic idea to use the monotonicity of the super-level sets will remain present.*

4.3 THE IDEA OF THE ALGORITHM

From the remark about the (theoretical) *optimal density* at the end of Section 3.2, to minimize the variance and hence the necessary number of simulation runs, we will strive for an algorithm that to some extend approximates the *ideal density* (3.30). We are trying to reach this goal by an

approximation with mixtures of Gaussian bell shaped curves centred around some *exploration points*. Thus, we will use some kind of *adapted importance sampling* consisting of the following two steps:

(i) A *preprocessing step* which corresponds to the exploration mentioned above. For a moderate (meaning not too large) value of α, we construct an algorithm to find points in the sample space $\mathbb{R}^d$ which cover in a sufficient way the set $\{h > \alpha\}$. The collection of these points is called *set of exploration points*. This initial step will only be carried out once.

(ii) Then, the *proper importance sampling step* is carried out for different values of α which are ordered in an increasing way. The purpose of this step twofold: On the one hand, the tail probability is estimated on the basis of an importance sampling distribution given by the mixture of Gaussian distributions centered at the exploration points. On the other hand, in the course of sampling, the set of exploration points will be constantly modified in favor of points where h attains larger values.

As said above, the basic observation is the monotonicity of the super - level sets which we are using to save running time by modifying the set of exploration points for $\alpha' \geq \alpha$ while we are estimating the tail probability for α at the same time.

Remark 6. *The use of Gaussian bell shaped curves for the approximation of the* ideal density *is completely arbitrary and mainly due to the fact that the density with respect to the true distribution of X can be easily calculated. More sophisticated choices are possible such as Gaussians with different variances or uniform distributions around balls of suitable diameters around the exploration points in order to look to tails $P(h(x) > a)$.*

4.4 THE PREPROCESSING STEP

The exploratory preprocessing step mentioned above is intended to gain some first information about the tail probabilities of $h(X)$, in particular some knowledge about points where h is large. The analysis depends on the level α chosen. Basically, we want to construct an algorithm which on the one hand finds exploration points in S_α but on the other hand also provides us with a criterion to stop the preprocessing when the set of exploration points is sufficiently large.

To choose the exploration points, we introduce three parameters that control the preprocessing, namely:

(i) The *exploration width* $\epsilon > 0$. The exploration width makes sure that the minimum distance between two exploration points is larger than $\epsilon > 0$ to control the number of exploration points.

(ii) The *exploration gain* $E > 1$. The exploration gain is a threshold parameter for the number of simulation runs allowed without changing the set of exploration points. That means, the preprocessing step is terminated after the set of exploration points has not changed in a suitable number E of consecutive simulation steps.

(iii) An *outback threshold* consisting of a fixed function $r : \mathbb{R}^d \to \mathbb{R}$ and some $B > 0$. The outback threshold helps to avoid to explore parts of the super - level set where the probability that X actually takes values in that part is orders of magnitude smaller than $\mathrm{P}(h > \alpha)$. It is imposed by discarding every sampled value $x \notin \mathcal{F}_B := \{x \,|\, r(x) < B\}$.

This requires some crude lower bound of the order of magnitude of p_α. The set $\mathcal{F}_B$ is called the *feasibility space* of the sampling method, the introduction of the outback threshold thus reduces the exploration of $\mathcal{S}_\alpha$ to the exploration of $\mathcal{S}_\alpha \cap \mathcal{F}_B$.

Remark 7. *(i) In the one - dimensional testbed below, we simply have* $r(x) := |x|$.
(ii) The proper choice of the parameters is absolutely crucial for the performance of the algorithm. To derive some criteria how to do that in an appropriate way is a difficult problem which so far can only be approached by extensive simulation studies which are beyond the scope of this short description.

To describe the exploration of the super - level sets in the course of the process, we have to keep track of the barycentres and the weights attached to the different Gaussian bell shaped curves. This requires some bookkeeping for which we introduce the notion of *exploration status*.

The exploration status

$$\mathcal{E}_\alpha(n) := (\Theta_1^{(n)}, \ldots, \Theta_{m_n}^{(n)} \,|\, w_1^{(n)}, ..., w_{m_n}^{(n)} \,|\, h(\Theta_1^{(n)}), \ldots, h(\Theta_{m_n}^{(n)}))$$

after the n-th exploration step is a collection of $1 \leq m_n \leq n$ *exploration points*

$$(\Theta_1^{(n)}, \ldots, \Theta_{m_n}^{(n)}),$$

$\Theta_i \in \mathbb{R}^d$, their values

$$(h(\Theta_1^{(n)}), \ldots, h(\Theta_{m_n}^{(n)})),$$

which are calculated and stored during the process and m_n *associated weights*

$$(w_1^{(n)}, ..., w_{m_n}^{(n)}),$$

where $w_i^{(n)} \geq 1$, $w_1^{(n)} + ... + w_{m_n}^{(n)} = n$ (thus $m_n \leq n$; m_n indicates the status length: it also reflects the number of local maximums observed sofar). The $w_j^{(n)}$ will count the number of times the point $\Theta_j^{(n)}$ was survivor in comparison with new points θ. It also indicates the width or size of the area around the local bump. Note that $p_j^{(n)} = w_j^{(n)}/n$ may serve as a probability density that allows to sample around extreme points.
The exploration status is obtained as follows:

1. For $n = 1$: The first exploration step is to sample some point θ for the distribution of X such that $h(\theta) > \alpha$ and $r\theta) \leq B$. This having done, the exploration status will be

   $$\mathcal{E}_\alpha(1) := (\Theta_1^{(1)} := \theta \,|\, w_1^{(1)} := 1 \,|\, h(\theta)).$$

   We set $m_1 = 1$. We also set $n_{\text{unchanged}} = 0$ (the number of times that the status length does not change).

2. In step $n+1$, we wait again until we sample some $\theta \in \mathbb{R}^d$ with $h(\theta) > \alpha$. A way to speed up this adaptively is decribed in the next subsection 4.5. Then, we consider the following alternative:

   (a) In case $\min \|\theta - \Theta_i^{(n)}\| > \epsilon$, while also $r(\theta) \leq B$ (a feasible point), we add the point θ to the exploration status, i.e. we set $m_{n+1} = m_n + 1$, $\Theta_{m_{n+1}}^{(n+1)} := \theta$, $w_{m_{n+1}}^{(n+1)} := 1$ and obtain for $\mathcal{E}_\alpha(n+1)$:

   $$(\Theta_1^{(n)}, \ldots, \Theta_{m_n}^{(n)}, \theta \,|\, w_1^{(n)}, \ldots, w_{m_n}^{(n)}, 1 |\, h(\Theta_1^{(n)}), \ldots, h(\Theta_{m_n}^{(n)}), h(\theta)).$$

   Thus $\Theta_{k'}^{(n+1)} = \Theta_{k'}^n$, $w_{k'}^{(n+1)} = w_{k'}^n$ for $k' \neq k$. We reset $n_{\text{unchanged}} = 0$.

(b) In case $\min \|\theta - \Theta_i^{(n)}\| \leq \epsilon$, we look for the exploration point which is closest to θ, i.e. let $\Theta_k^{(n)}$ be this point. Now we have two alternatives:

- **reject** θ**:** If $r(\theta) > B$ (point θ is unfeasible), or $h(\Theta_k^{(n)}) > h(\theta)$ (old point $\Theta_k^{(n)}$ is better then θ), the exploration status remains unchanged ($n_{\text{unchanged}} = n_{\text{unchanged}} + 1$) except that now $w_k^{(n+1)} = w_k^{(n)} + 1$ is increased by one (Θ_k^n is a survivor). Thus $\Theta_{k'}^{(n+1)} = \Theta_{k'}^n$ (for all k'), $w_{k'}^{(n+1)} = w_{k'}^n$ for $k' \neq k$. Set $m_{n+1} = m_n$.
- **accept** θ**:** If $h(\Theta_k^{(n)}) \leq h(\theta)$, then θ will replace Θ_k^n and will inherit the qualifications of the last in the exploration status. Thus the status length remains unchanged except the modification $\Theta_k^{(n+1)} = \theta$, $h(\Theta_k^{(n+1)}) = h(\theta)$, $w_k^{(n+1)} = w_k^{(n)} + 1$ and $\Theta_{k'}^{(n+1)} = \Theta_{k'}^n$, $w_{k'}^{(n+1)} = w_{k'}^n$ for $k' \neq k$. Set $m_{n+1} = m_n$. Also in this case we decide to increase $n_{\text{unchanged}} = n_{\text{unchanged}} + 1$.

3. The *stopping criterion*: We will stop the exploration if the vector $(\Theta_1^{(n)}, \ldots, \Theta_{m_n}^{(n)})$ of exploration points status remains constant for $n_{\text{unchanged}} \geq E$.

4.5 THE PROPER IMPORTANCE SAMPLING STEP

After the exploration step for $\alpha > 0$, we have a vector $(\Theta_1^{(n)}, \ldots, \Theta_{m_n}^{(n)})$ of exploration points together with an associated weight vector $(w_1^{(n)}, \ldots, w_{m_n}^{(n)})$. These exploration points are constructed to cover the super - level set $\mathcal{S}_\alpha$ in a way that is sufficiently accurate for the subsequent important sampling from the mixture distribution. The mixture distribution associated to a given exploration status is obtained by centering the standard normal distribution given by the density ρ above around the exploration points $\Theta_i^{(n)}$, weighting them with the values $p_i = \frac{w_i}{n}$ (note that $\sum_{i=1}^{m_n} p_i = 1$). Thus, our new sampling distribution is given by the mixture density

$$\pi_n(x) = \sum_{i=1}^{m_n} p_i^{(n)} \frac{\rho(x)}{\rho(x - \Theta_i^{(n)})} , \tag{4.1}$$

where the ratio can be evaluated by expanding the square in the exponent of the normal distribution. The approximation of the *optimal sampling density* (3.30) after n exploration runs is then given by

$$\tilde{\pi}_n(x) = I_{\{h(x)>\alpha\}} \sum_{k=1}^{n} p_i^{(n)} \exp\left\{ \frac{1}{2} \Theta_{i(k)}^{(n)\,T} \Sigma^{-1} \Theta_{i(k)}^{(n)} - x^T \Sigma^{-1} \Theta_{i(k)}^{(n)} \right\}$$

[we assumed the same Σ-matrix in both ρ densities], where $i(k)$ indicates a subset of $1, \ldots, n$ (only m_n indices will be used).

As already said above, it is natural to expect that it is more efficient to look for exploration points of $\mathcal{S}_{\alpha'}$, $\alpha' > \alpha$, on the basis of the information already given by the exploration status $\mathcal{E}_\alpha(n)$ for α after n exploration runs rather than to use the same exploration algorithm again. Thus, the sampling is *adaptive* in the sense that the exploration status and therefore also the mixture distribution (4.1) that we use for the sampling, may change in the course of the procedure.

Due to the fact that the super-level sets decrease, we may even want to remove exploration points

from $\mathcal{E}_\alpha(n)$. There are certainly many different reasonable ways to do that. We use the following approach where we sample in the first place from a distribution which may be more concentrated around its barycentre than those appearing in the mixture distribution.

We think of $\mathcal{E}_\alpha(n)$ as the *seed status*.

- For each $\Theta_i^{(n)}$, let $\omega = w_i^{(n)}$. We assume that $h(\Theta_i^{(n)})$ is known.
  - We sample $\omega \times E$ times from a normal variable as follows:
    (a) Draw a value θ from a $N(\Theta_i^{(n)}, \Sigma')$-variable where $\Sigma' = a\Sigma$ and $a \leq 1$. Check feasibility ($f(\theta) \leq B$) and evaluate $h(\theta)$.
    (b) If $r(\theta) \leq B$ we have three alternatives:
      - if $h(\theta) > h(\Theta_i^{(n)})$ set for the next draw $\Theta^{(n)} := \theta$, $h(\Theta_i^{(n)}) := h(\theta)$ and $w_i^{(n)} := w_i^{(n)} + 1$ and proceed to step (a),
      - if $\alpha' \leq h(\theta) \leq h(\Theta_i^{(n)})$ just change the status to $w_i^{(n)} := w_i^{(n)} + 1$ and proceed to step (a),
      - if $\alpha' > h(\theta)$ just proceed with step (a),

    (c) Stop when we have drawn $\omega \times E$ times.
  - If after the simulations for drawing a θ, we have $h(\Theta_i^{(n)}) > \alpha'$, then we transfer $\Theta_i^{(n)}$, $h(\Theta_i^{(n)})$ and $w_i^{(n)}$ to the new exploration status. Otherwise, the point $\Theta_i^{(n)}$ *is no longer considered*.

Collecting location, weights and function value from points with $h(\Theta_i^{(n)}) > \alpha'$, we obtain the new exploration status $\mathcal{E}_{\alpha'}(n')$ where n' is the sum of the final weights of all exploration points which were not removed in step (c). Thus, the new exploration status consists at most of as many exploration points as in $\mathcal{E}_\alpha(n)$ with different weights.

4.6 DISCUSSION AND OUTLOOK

The basic idea of the proposed adaptive importance sampling algorithm can be found at many places in the literature (f.i. [27]). Due to our assumptions about the structure of the problem, we propose an adaptive approach to construct the exploration points by a rather time consuming *exploration step* which has to be performed only once. The hope is that this disadvantage is outbalanced by the flexibility that is now gained by a considerable improvement in the sampling step and in the exploration of the super-level set $\mathcal{S}_\alpha$ for increasing values of α.

The way to achieve this is by no means unique. Different sampling distributions could be used, different ways to explore the function, and so on. But after all, it is not expected that there is one optimal way to resolve the problem stated above with a miraculous gain in sampling efficiency. At one point one will always be confronted with the fact that calculating the function is costly. This holds no longer true if some of our basic assumptions about the problem do not hold any longer, for instance if there appears a way to calculate h in a fast way. Then we might have to think about some completely different algorithm.

Three further refinements one could think of are:

- Adaption of the variance for the sampling around a given exploration point using smaller variances for points with large weights.

- Introduction of an *outback threshold* also in the step, where the super-level set for $\alpha' > \alpha$ is explored. That would help to avoid rather improbable values for θ (cf. the problems in sampling an *outback-phantom* below).

- An adaptive approach also for the preprocessing step where so far is always sampled from the raw distribution of X.

The performance of the scheme depends heavily on the quality of the exploration in the preprocessing step and hence on the choice of the three control parameters. However, this and therefore the performance of the whole adapted importance sampling algorithm is notoriously difficult to evaluate on a theoretical base. To get some intuition how to choose the correct parameter values for a given application thus requires extensive simulation studies. It seems to be reasonable to carry out the first exploration step as accurately as possible. For that, the starting value for α should be moderate (such that p_α is not too small) and the exploration gain E should be chosen rather large (such that there are many exploration points in this first approach to the function). But we also have to be aware of the fact that the algorithm will perform better for smooth functions h and may not work at all if h is very rough, for instance if there are a number of small spikes. Another problem is to choose the step size when increasing the values for α. If the difference between α and the next value α' is too large, the removal of exploration points described above is likely to become unreliable. The problem of choosing the step size seems therefore similar to the problem to find a suitable cooling schedule in simulated annealing. All this has to be investigated.

Another problem is the reliability of the results produced by the algorithm. For the importance sampling from the mixing distribution, we can only give bounds on the number of runs necessary for a given accuracy on the basis of the (unknown) variance of the sampling variable. It might therefore be necessary to use *convergence diagnostics* (cf. for instance the survey [10]) to find suitable stopping criteria for the actual importance sampling.

4.7 A 1-D-TESTBED

Finally, we will consider some simple simulations of the algorithm in a one-dimensional testbed. Note that this should be seen just as some first approach to demonstrate that the algorithm basically works. Even though the situation in one dimension is certainly simpler than in general, we think that this is sufficient for a first judgement since the performance of MC - methods is commonly believed not to depend too much on the dimension d of the sample space (cf. for instance the paragraph on Monte - Carlo methods in [53], p. 30 ff.).

We investigate the algorithm (see Appendix B for the source code) for real functions of a single standard normal random variable X. Testbeds are real functions $h : \mathbb{R} \to \mathbb{R}$ consisting of mixtures of three different types representing three different qualitative ways how large values of f may occur, namely:

(i) The *bump-phantom*: $\phi_i(x) := a_i(1 - ((x - x_i)/b_i)^2)$, $a_i, b_i > 0$, $x_i \in \mathbb{R}$: Small contributions to the tail probabilities p_α arise from those parts of a function, if α is slightly smaller than a. But bump phantoms are also important to test the exploration part of the algorithm. If $\alpha < a$ is changed to some $\alpha' > a$, all exploration points in $\mathcal{E}_\alpha$ which explore this particular bump should be removed while proceeding to $\mathcal{E}_{\alpha'}$.

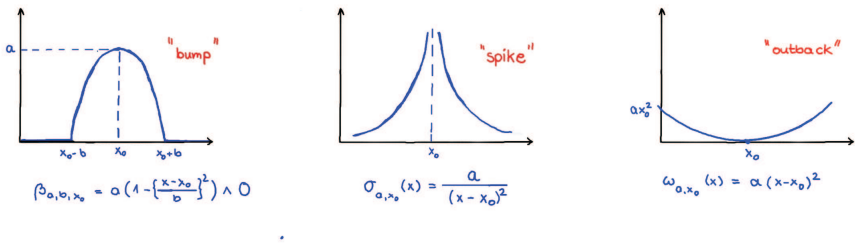

Figure 4.1: The "bump"-, "spike"- and "outback"-phantoms

(ii) The *spike-phantom*: $\psi_k(x) := \frac{a_k}{(x-x_k)^2}$, $a_k > 0$, $x_k \in \mathbb{R}$: This phantom represents singularities where arbitrary large values occur. In creasing the level α, the exploration should yield less and less points which come closer and closer to the actual location of the spike.

(iii) The *outback-phantom*: $\chi_j(x) := a_j(x - x_j)^2$, $a > 0$, $x_j \in \mathbb{R}$: This phantom represents large function values which are located close to the outback threshold and are therefore explored with only a small probability. Those parts of the super - level set caused by outback - phantoms seem difficult to explore and require particular large values for the exploration gain parameter E in order to be discovered.

To test for possible interactive effects between these basic types we can combine these basic phantoms to more general testbed functions h given by

$$h(x) := \max\{\phi_1(x), ..., \phi_l(x), \chi_1(x), ..., \chi_m(x), \psi_1(x), ..., \psi_n(x), 0\}$$

but this is only considered briefly at the end of this first approach. We choose the *outback threshold* of the feasibility set defined by $r(x) = |x|$ to be $B = 8.5$ since for a standard normal variable, we have

$$P(|X| > 8.5) = 2 \times \Phi(-8.5) = 1.9 \times 10^{-17}.$$

To judge about the performance of the algorithm, we calculate first some true tail probabilities for the three phantoms. Note that these probabilities are exactly calculated using the R-implementation of the distribution function of a standard normal variable. That we can do this was the major reason to restrict ourselves to tests in dimension one.

- In the first case, the *bump phantoms* are given by

  $$\phi_A(x) = 100(1 - A(x - 4)^2),$$

  with for A values $A_i = 10, 100, 1000$ (labelling the rows) and the table displays the probabilities $P_{ij} = P(\phi_{A_i} > \alpha_j)$ with $\alpha_j = 5, 10, 15$ (columns).

A_i		$P_{ij} = P(\phi_{A_i} > \alpha_j)$		
		α_1=5	α_2=10	α_3=15
A_1=	10	1.031167e-04	9.925983e-05	9.539331e-05
A_2=	100	2.671111e-05	**2.596663e-05**	2.520389e-05
A_3=	1000	8.269453e-06	8.047889e-06	7.820166e-06

[$P_{2\,2}$ is made boldface for future reference]

- For the *spike phantoms*

$$\psi_A(x) = \frac{1}{(x-A)^2}$$

we determine the probabilities $P_{ij} = P(\psi_{A_i} > \alpha_j)$ for $A_i = 2, 5, 7$ and $\alpha_j = 5, 10, 15$.

A_i	$P_{ij} = P(\phi_{A_i} > \alpha_j)$		
	α_1=5	α_2=10	α_3=15
A_1= 2	5.303881e-02	3.583993e-02	2.880499e-02
A_2= 5	2.621418e-06	**1.355245e-06**	9.863961e-07
A_3= 7	2.818901e-11	1.151590e-11	7.625567e-12

[$P_{2\,2}$ is made boldface for future reference]

- Finally, for the *outback-phantoms*

$$\chi_A(x) = A\,x^2$$

we determine the probabilities $P_{ij} = P(\chi_{A_i} > \alpha_j)$ for $A_i = 0.75, 0.5, 0.25$ and $\alpha_j = 5, 10, 15$.

A_i	$P_{ij} = P(\phi_{A_i} > \alpha_j)$		
	α_1=5	α_2=10	α_3=15
A_1= 0.75	9.823275e-03	2.607296e-04	7.744216e-06
A_2= 0.50	1.565402e-03	**7.744216e-06**	4.320463e-08
A_3= 0.25	7.744216e-06	2.539629e-10	9.485738e-15

[$P_{2\,2}$ is made boldface for future reference] Compared to these true tail probabilities, the outback threshold is properly chosen.

We choose now one particular case for each example to estimate a tail probability by our algorithm (see Appendix B for the functions EFS, WS and "c"). Please note that the EFS-step might give an output "NULL" which means that in that case, the alpha-value was chosen too small in the exploration step in relation to the alpha-value in EFS.

1. We choose the bump phantom given by

$$\phi_A(x) = 100(1 - A(x-4)^2),$$

with for $A = 100$. We are interested in obtaining $P = P(\phi_A > \alpha)$ for $\alpha = 10$. Note that this corresponds to $P = P_{2\,2} = 2.596663e-05$. We do this adaptively in two steps. First we determine the exploration status for the level $\alpha = 2$ and then increase the level to $\alpha' = 10$.

   - The first exploration step (with level $\alpha = 2$) yields the exploration status (θ, w).

```
> status
[1] 3.990454 5.000000
```

   - The two subsequent steps with new level $\alpha = 10$ and previous weight $w = 5$ yield the updated exploration status


```
> EFS(status,10,5)
[1] 3.995818 6.000000
```

- To discuss the performance of the algorithm, we now aim to estimate the 0.95-quantile of the absolute difference between the true probability and the simulation for importance sampling with $N_{IS} = 40000$ runs on the basis of 20 subsequent simulations. Steps:
  - Call WS$(\theta, w, \alpha, N_{IS})$.
  - Determine relative error when compared to $P = P_{2\,2} = 2.596663e-05$.

```
> out<-c();
  for(k in 1:20){out[k]<-WS(3.995818,6,10,40000)};
  dif <- abs((out - 0.0000259)/0.0000259);
  print(dif[0.95*20])
  [1] 0.001446600
```

  We thus obtain that with probability 0.95, the relative difference Θ is of order 10^{-3} after $N_{IS} = 40000$ importance sampling steps. The theoretical value for naive Monte Carlo from the Cramér bound is given by approximately $N_{MC} = 2.3 \times 10^{11}$.

Thus, the sampling points θ found during the process are reasonably close to the single bump at $x = 4$ and the tail probability is reasonably close to the true $p = 2.597 \times 10^{-5}$ from the exact calculation above.

2. For the *spike phantoms*

$$\psi_A(x) = \frac{1}{(x-A)^2}$$

we take $A = 5$. We are interested in obtaining $P = P(\phi_A > \alpha)$ for $\alpha = 10$. Note that this corresponds to $P = P_{2\,2} = 1.355245e-06$. We do this again adaptively in two steps. First we determine the exploration status for the level $\alpha = 2$ and $E = 5$, then increase the level to $\alpha' = 10$.

- The first exploration step (with level $\alpha = 2$) yields the exploration status (θ, w).

```
> status
[1] 4.775324 5.000000
```

- The adaptive step with new level $\alpha = 10$ and previous weight $w = 5$ yield the updated exploration status

```
> EFS(status,10,5)
[1] 5.259688 6.000000
```

- In order to consider the performance of the algorithm, we estimate the 0.95-quantile of the absolute difference between the true probability and the simulation for importance sampling with $N_{IS} = 40000$ runs on the basis of 20 subsequent simulations. Steps:
  - Apply the weighted sampling WS$(\theta, w, \alpha, N_{IS})$.
  - Determine relative error when compared to $P = P_{2\,2} = 1.355245e-06$.

```
> out<-c();
  for(k in 1:20){out[k]<-WS(5.259688,6,10,40000)};
  dif <- abs((out - 0.000001355)/0.000001355);
  print(dif[0.95*20])
[1] 0.004649013
```

This yields a 0.95-quantile for the relative error of order 10^{-3}. The required number of steps for 'naive' (normal MC) sampling necessary to achieve a comparative accuracy is of the same order of magnitude as in the case of the bump.

Again, the sampling points θ found during the process are reasonably close to the single spike at $x = 5$ and the tail probability is reasonably close to the true $p = 1.355 \times 10^{-6}$ from the exact calculation above.

3. For the *outback phantom*

$$\chi_A(x) = A\,x^2$$

we take $A = 0.5$. We are interested in obtaining $P = P(\phi_A > \alpha)$ for $\alpha = 10$. Note that this corresponds to $P = P_{2\,2} = 7.744216e-06$. We do this again adaptively in two steps. First we determine the exploration status for the level $\alpha = 2$ and $E = 5$, then increase the level to $\alpha' = 10$.

- The first exploration step (with level $\alpha = 2$ and $n = 3$) yields the exploration status $(\theta_1, \theta_2, \theta_3, w_1, w_2, w_3)$.

```
> status
[1]  3.943461  4.468545 -3.795828  7.000000  1.000000
7.000000
```

- Next, increasing the exploration level to $\alpha = 10$ and stopping factor $E = 5$ now yields for the exploration-from-seed functionality

```
EFS(status,10,5)
[1]  9.667335  4.880202 -6.971651 35.000000  4.000000
33.000000
```

- Here the results differ largely from $P = P_{2\,2} = 7.744216e-06$. Due to the problems with the outback function we only consider single simulations varying the respective exploration points.
  We apply the weighted sampling WS($c(\theta_1, \theta_2, \theta_3)$, $c(w_1, w_2, w_3)$, α, N_{IS})

```
> WS(c(9.667335,4.880202,-6.971651),c(35,4,33),10,5000)
[1] 1.808273e-06
> WS(c(4.880202,-6.971651),c(4,33),10,5000)
[1] 3.819555e-06
> WS(c(4.880202),c(4),10,5000)
[1] 3.595187e-06
```

which shows that a size restriction would be good for the exploration-from-seed step as well. In total, we see that the sampling for these outback phantoms does not perform as good as for the others.

The true probability $P = P_{2\,2} = 7.74 \times 10^{-6}$ deviates considerably from the estimate but is at least of the true order of magnitude. Probably the three sampling points do not sufficiently cover $\mathcal{S}_\alpha$ and one should use a larger value for the exploration gain to overcome this difficulty. Clearly that will increase the duration of the preprocessing step and we have to keep an eye on that.

4. Finally, for a combined function $h(x) = \psi_{A=-4}(x) + \chi_{A=0.5}(x) + \phi_{A=100}(x)$

   ```
   h <-function(t){x<-1/(t+4)^2+0.5*t^2+100*(1-100*(t-4)^2)},
   ```

   we get setting the exploration point distance $\varepsilon = 2$ and the stopping factor $E = 200$ in the program macro *EXPLORATION*

   - The first exploration step yields the exploration status $(\theta_1, \theta_2, w_1, w_2)$

     ```
     > status
     [1]   4.000018  -3.999809 397.000000   3.000000
     ```

   - By the exploration-from-seed this is upgarded to

     ```
     > EFS(status,10,5)
     [1]   4.000235  -3.999809 702.000000   3.000000
     ```

   - The weighted sampling WS($c(\theta_1, \theta_2),\ c(w_1, w_2),\ \alpha, N_{IS}$)

     ```
     > WS(c(4.000235,-3.999809),c(702,3),10, ...)
     [1] 2.633854e-05
     ```

   We observed that it is important to use a very large value for the stopping factor E in order to find all different places with large values of h, in particular if the probabilities for these regions differ by orders of magnitude.

Section 5

Prototype procedure Importance Sampling

Static RAM (SRAM) performance can be described by evaluating response functions like Static Noise Margin, Write Margin, Read Current, and Leakage Current as function of several parameters. Batches of wafers of chips and a series of dies on each wafer (each die containing SRAMs are subject to process variations. One considers inter-die process variations (that are correlated) and intra-die ones (that are stochastic). Variability is limiting SRAM performance: one has to be able to distinguish between a writable (unstable) memory cell and readability without flipping situation (stable). Due to technology scaling one has to deal with an increased number of bits and an increased process spread. Variability has always been an important issue for SRAM. In the past an additional design margin was taken into account to ensure the memory would operate even though the distributions were approximated by some extrapolation technique (like we will check in Section 5.3). Nowadays it is not possible anymore to use additional design margins if one wants to continue technology scaling. The increased number of bits in combination with the increased variability has left very little margin. We can even say that variability has become critical for SRAM performance. Therefore an accurate estimation of the tails of the distributions has become important.
Each SRAM memory is composed of several transistors. The same transistors are used for reading and writing. Hence, important input parameters for the response functions are transistor parameters like V_T (threshold voltage) and β (current amplification factor).

In this section we describe a prototype procedure used to apply Importance Sampling. This prototype was used in simulating results presented in the forthcoming Section 6. The actual code can be found in Appendix A. As output quantity we consider the Static Noise Margin $\mathrm{SNM}(V_{T,1}, \ldots, V_{T,6})$, where each of the input parameters $V_{T,j}$ is taken according to some density function. The Static Noise Margin function does not show a normal density distribution. Hence we will not assume this in the current Section. Note that Section 6 will a way to circumvent this for this particular function.
The notions of f-distribution for the original distribution in the parameter space and of g-distribution of the one used by importance sampling will be similar as used earlier in Section 3. In Section 5.1 we will sample the $V_{T,j}$ from a broad uniform distribution, while in Section 5.2 we will sample according to the normal distribution $V_{T,j} \approx \mathrm{N}(\mu^{(j)}, \sigma^{(j)})$ (i.e. the f-distribution) as for standard Monte Carlo. Section 5.3 will demonstrate that the "Extrapolated Monte Carlo" approach erroneously under-estimates the cumulative density function of this particular SNM

distribution.
For calculating the Static Noise Margin as postprocessing facility on the results of a circuit simulation we refer to [9, 14].
We refer to Appendix A for the Matlab code that is associated with this section.

5.1 IMPORTANCE SAMPLING MONTE CARLO

To apply *Importance Sampling MC* we sample the $V_{T,j}$ by a broad uniform distribution, $v_{T,j} \sim \mathrm{Unif}(\mu, \kappa\sigma)$, with $\kappa = 6$ (i.e. the g-distribution). In this way we get N sampled tuples $\mathbf{v}^{(k)} = (V_{T,1}^{(k)} \ldots, V_{T,6}^{(k)})$, $k = 1, \ldots N = 10^5$. Note that this is much less than the 10^{12} samples mentioned at (2.36).
Let $\mathrm{M}^{(j)} = \max_k(V_{T,j}^{(k)})$ and $\mathrm{m}^{(j)} = \min_k(V_{T,j}^{(k)})$.
First we estimate for each j-th parameter $\mu^{(j)}$ by $\hat{\mu}^{(j)} := \mathrm{mean}_k(V_{T,j}^{(k)})$ and $\sigma^{(j)}$ by $\hat{\sigma}^{(j)} := (\mathrm{M}^{(j)} - \mu^{(j)})/\kappa$. From this we define the individual approximative densities

$$f_j(x) = \frac{1}{\sqrt{2\pi}} \frac{1}{\hat{\sigma}^{(j)}} e^{-\frac{1}{2}(\frac{x-\hat{\mu}^{(j)}}{\hat{\sigma}^{(j)}})^2}. \tag{5.1}$$

With this the multi-parameter distribution f is obtained

$$f(\mathbf{v}) = f(V_{T,1} \ldots, V_{T,6}) = \prod_{j=1}^{6} f_j(V_{T,j}). \tag{5.2}$$

Next, with the parameter range length $\mathrm{R}^{(j)} = \mathrm{M}^{(j)} - \mathrm{m}^{(j)}$ we estimate each (uniform) g_j-density distribution by

$$g_j(x) \quad = \quad \begin{cases} \frac{1}{\mathrm{R}^{(j)}} & \text{if} \quad x \in [\mathrm{m}^{(j)}, \mathrm{M}^{(j)}] \\ 0 & \text{else} \end{cases} \quad . \tag{5.3}$$

With this the multi-parameter distribution g becomes

$$g(\mathbf{v}) = g(V_{T,1} \ldots, V_{T,6}) = \prod_{j=1}^{6} g_j(V_{T,j}). \tag{5.4}$$

The f/g ratio now becomes $\phi(\mathbf{v}) = \phi(V_{T,1} \ldots, V_{T,6}) = \prod_{j=1}^{6} \frac{f_j(V_{T,j})}{g_j(V_{T,j})}$.
For the cumulative probability function $\mathrm{cdf}_{\mathrm{SNM}}(X) = P(\mathrm{SNM}(\mathbf{v}) \leq X)$ we have to determine

$$\mathrm{cdf}_{\mathrm{SNM}}(X) = P(\mathrm{SNM}(\mathbf{v}) \leq X) = \frac{1}{N} \sum_{k}^{N} I_{\mathrm{SNM}(\mathbf{v}^{(k)}) \leq X} \; \phi(\mathbf{v}^{(k)}). \tag{5.5}$$

To approximate the cumulative probability function we determine a histogram. Let

$$\mathrm{M}_{\mathrm{SNM}} \quad = \quad \max_k(\mathrm{SNM}(V_{T,1}^{(k)} \ldots, V_{T,6}^{(k)})), \tag{5.6}$$

$$\mathrm{m}_{\mathrm{SNM}} \quad = \quad \min_k(\mathrm{SNM}(V_{T,1}^{(k)} \ldots, V_{T,6}^{(k)})). \tag{5.7}$$

Then the range length of the values of the output function is defined by $\mathrm{R}_{\mathrm{SNM}} = \mathrm{M}_{\mathrm{SNM}} - \mathrm{m}_{\mathrm{SNM}}$.
Let $X_{\mathrm{SNM}}[i] = \mathrm{m}_{\mathrm{SNM}} + (i-1) * \mathrm{R}_{\mathrm{SNM}}/s$ be the i-th bin bound, where $s = 250$ (bin size) and

$i = 1, \ldots s+1$. Hence the i-th bin is defined by $(X_{\rm SNM}[i], X_{\rm SNM}[i+1]]$, $i = 2, \ldots s$, while the first interval just is the single point $\{X_{\rm SNM}[1]\}$. Define the bin probability function by

$$f_{b,\rm ISMC}[i] \;=\; \begin{cases} \mathrm{P}(X_{\rm SNM} \in (X_{\rm SNM}[i], X_{\rm SNM}[i+1]]) & \text{if } \; i > 1 \\ \mathrm{P}(X_{\rm SNM} = X_{\rm SNM}[1]) & \text{if } \; i = 1 \end{cases}, \tag{5.8}$$

in which $\mathrm{P}(X_{\rm SNM} \in (X_{\rm SNM}[i], X_{\rm SNM}[i+1]])$ is determined by determining the relative occurrence of the outcomes of the SNM function in this interval, weighted by the f/g ratio. Now the cumulative probability function can be approximated by

$$\mathrm{P}_{\rm ISMC}(X^{\rm unif}_{\rm SNM} \leq X_{\rm SNM}[i]) \;\approx\; \sum_{m=1}^{i} f_{b,\rm ISMC}[m]. \tag{5.9}$$

We make several remarks

- After ordering the SNM-values in increasing order the accuracy of the cumulative probability function can be improved by applying weighted Trapezoidal Rule quadrature on f on successive ordered outcomes $X^{(a)}_{\rm SNM} \leq X^{(b)}_{\rm SNM}$. We indicate the corresponding samples k by k_a and k_b, respectively. Then

  $$\frac{X^{(b)}_{\rm SNM} - X^{(a)}_{\rm SNM}}{2\,\mathrm{R}_{\rm SNM}}\,[P(X_{\rm SNM} = X^{(a)}_{\rm SNM}) + P(X_{\rm SNM} = X^{(b)}_{\rm SNM})] \tag{5.10}$$

  In doing this we assumed to have filtered out multiple occurrences and to have incorporated their effect in the chances $P()$ at the right-hand side of (5.10). Note that

  $$P(X_{\rm SNM} = X_{\rm SNM^{(a)}}) = \frac{1}{N}\sum_{k}^{N} I_{\rm SNM(\mathbf{v}^{(k)}=SNM(\mathbf{v}^{(a)})}\;\phi(\mathbf{v}^{(k)}) \tag{5.11}$$

  (note that these ϕ-values may be different for different k associated with function results with the same $\mathrm{SNM}(\mathbf{v}^{(a)})$ value. There may be even very large variations.

- When dealing with parameters in a multidimensional parameter space the sensitivity with respect to the parameters may be taken into account.

5.2 STANDARD MONTE CARLO

Similarly to the above, for a *Standard Monte Carlo* we look at the output function $\Sigma^{\rm norm}_{\rm SNM} = \mathrm{SNM}(\widetilde{V}_{T,1} \ldots, \widetilde{V}_{T,6})$, where the $\widetilde{V}_{T,j}$ are sampled according to a Normal distribution $\widetilde{V}_{T,j} \sim \mathrm{N}(\mu, \sigma)$, resulting in tuples $(\widetilde{V}^{(k)}_{T,1} \ldots, \widetilde{V}^{(k)}_{T,6})$, $k = 1, \ldots N = 10^5$. Let

$$\widetilde{\mathrm{M}}_{\rm SNM} \;=\; \max_k(\mathrm{SNM}(\widetilde{V}^{(k)}_{T,1} \ldots, \widetilde{V}^{(k)}_{T,6})), \tag{5.12}$$

$$\widetilde{\mathrm{m}}_{\rm SNM} \;=\; \min_k(\mathrm{SNM}(\widetilde{V}^{(k)}_{T,1} \ldots, \widetilde{V}^{(k)}_{T,6})). \tag{5.13}$$

Then the range length of the values is defined by $\widetilde{\mathrm{R}}_{\rm SNM} = \widetilde{\mathrm{M}}_{\rm SNM} - \widetilde{\mathrm{m}}_{\rm SNM}$. Let $\widetilde{X}_{\rm SNM}[i] = \widetilde{\mathrm{m}}_{\rm SNM} + (i-1) * \widetilde{\mathrm{R}}_{\rm SNM}/s$ again be the i-th bin bound, where $s = 250$ (bin size) and $i = 1, \ldots (s+1)$. The bin probability function is defined similarly as in (5.8) by sampling the relative occurrences of the output function in this interval (i.e. actually weighted by the f/g ratio equal to 1). The cumulative probability function is derived by a histogram (bin size equal to the one used as in the Importance Sampling case).

5.3 EXTRAPOLATED MONTE CARLO

For *Extrapolated MC* we boldly assume that the output density function $f_{\rm SNM}(s)$ is normal. We estimate $\mu_{\rm SNM}$ and $\sigma_{\rm SNM}$ by

$$\hat{\mu}_{\rm SNM} \approx {\rm mean}_k({\rm SNM}(\widetilde{V}_{T,1}^{(k)} \ldots, \widetilde{V}_{T,6}^{(k)})), \tag{5.14}$$

$$\hat{\sigma}_{\rm SNM} \approx {\rm std}_k({\rm SNM}(\widetilde{V}_{T,1}^{(k)} \ldots, \widetilde{V}_{T,6}^{(k)})). \tag{5.15}$$

Let $\hat{X}_{\rm snm}[i] = {\rm m_{SNM}} + (i - \frac{1}{2}) * {\rm R_{SNM}}/s$ be the i-th bin center, where $s = 250$ (bin size) and $i = 1, \ldots s$. The bin probability function is defined by

$$f_{b,{\rm EXMC}}[i] = \frac{1}{\sqrt{2\pi}} \frac{1}{\hat{\sigma}_{\rm SNM}} \exp[-\frac{1}{2}(\frac{\hat{X}_{\rm snm}[i] - \hat{\mu}_{\rm SNM}}{\hat{\sigma}_{\rm SNM}})^2] \times {\rm R_{SNM}}/s, \tag{5.16}$$

from which again a cumulative probability function can be derived. In this case the last function could also have been determined exactly.

5.4 COMPARISONS

In Figure 5.1 the cumulative probability functions (CPFs) obtained by MC using Importance Sampling, by Standard MC, and by 'Extrapolated MC' are shown. Clearly the CPF of the 'Extrapolated MC' deviates already quite soon from the other two CPFs due to the non-normality of the distribution of the output function SNM. There even is a consequent under estimation. The CPFs of the Normal MC and of the Importance Sampling MC are consistent for $10^{-5} \leq {\rm P}(x < X)$. Clearly, Importance Sampling MC is able to continue to even below 10^{-15}.

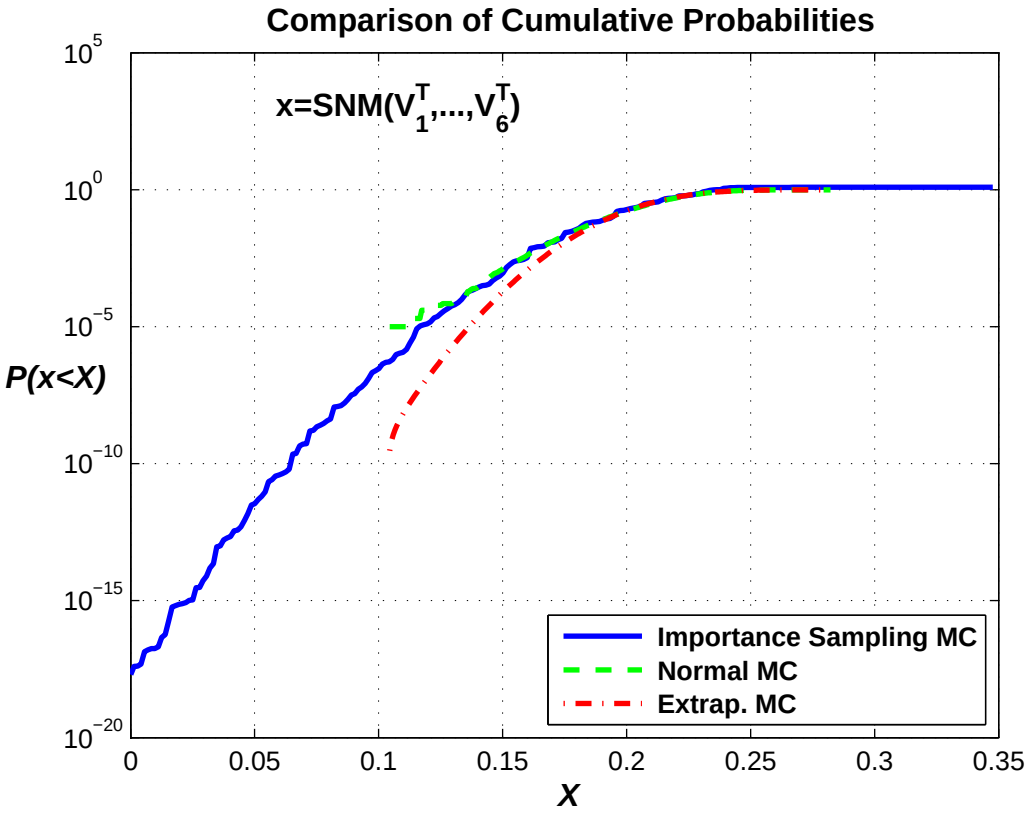

Figure 5.1: Cumulative probability functions by MC using Importance Sampling, by Standard MC, and by 'Extrapolated MC'

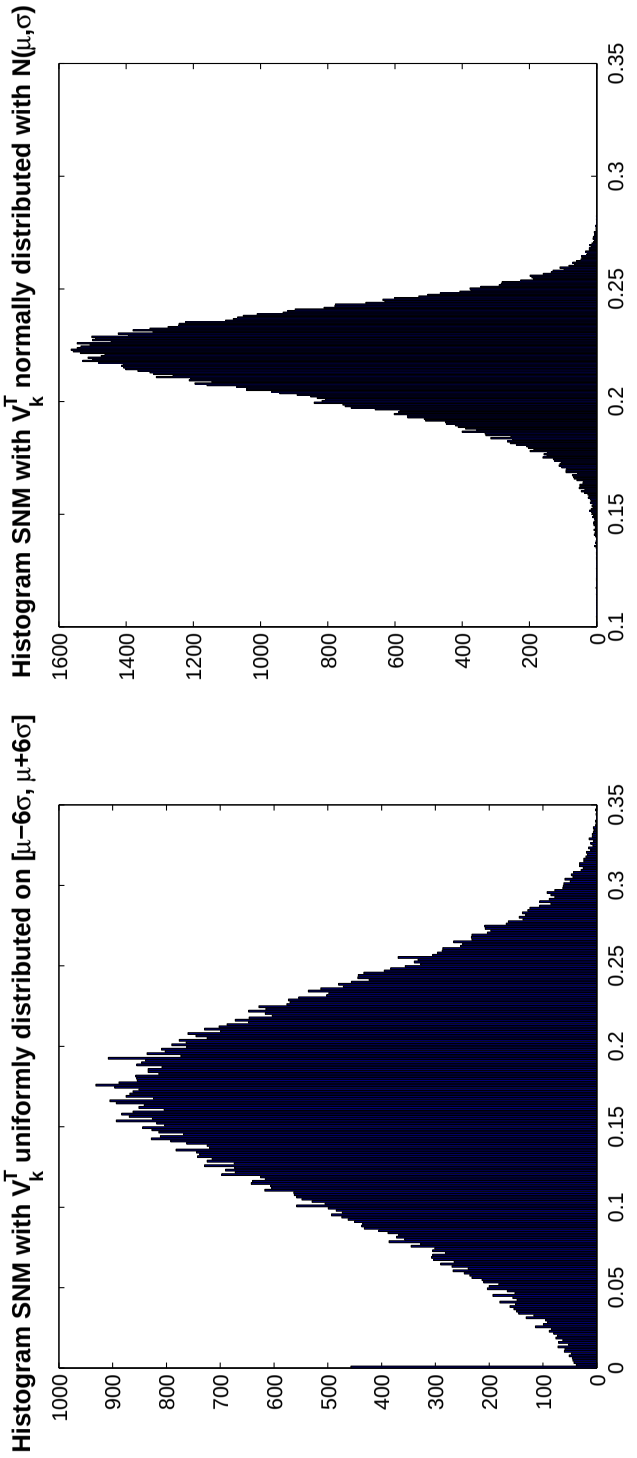

Figure 5.2: Histograms of SNM using v_T-s by Importance Sampling (left), and by Normal distribution (right)'

Section 6

Importance Sampling Monte Carlo Simulations for Accurate Estimation of SRAM Yield

6.1 ABSTRACT

[1]Variability is an important aspect of SRAM cell design. Failure probabilities of $P_{\text{fail}} \leq 10^{-10}$ have to be estimated through statistical simulations. Accurate statistical techniques such as Importance Sampling Monte Carlo simulations are essential to accurately and efficiently estimate such low failure probabilities. This chapter shows that a simple form of Importance Sampling is sufficient for simulating $P_{\text{fail}} \leq 10^{-10}$ for the SRAM parameters *Static Noise Margin* (SNM), *Write Margin* (WM) and *Read Current*. For the SNM, a new simple technique is proposed that allows extrapolating the SNM distribution based on a limited number of trials. For SRAM *Total Leakage Currents*, it suffices to take the averages into account for designing SRAM cells and modules. A guideline is proposed to ensure Bitline Leakage Currents do not compromise SRAM functionality.

6.2 INTRODUCTION

Decades of scaling according to Moores law have shrunk devices to such an extent that variability has become a serious issue at all levels of circuit design. The effects of variability are most noticeable in SRAM design, since SRAM cells use very small transistors. For this reason, statistics have long been part of SRAM cell design. Intra-die transistor V_t mismatch is still the main statistical parameter, although others are gaining importance. Downscaling of transistors leads to widened V_t-distributions (Figure 6.1-left). In addition, the amount of SRAM on large System-on-Chips (SoC's) continues to increase, causing the amount of variation that has to be taken into account to increase as well (Figure 6.1-right).
On top of this, there is a clear trend towards voltage scalable systems [9, 38], resulting in an increased demand for voltage scalable SRAM as well. At lower supply voltages, SRAM's are more susceptible to variability, leaving less design margin for the designer. Hence it is becoming

[1]This chapter was presented at the ESSCIRC 2008 Conference in Edinburgh, Scotland, Sept. 19, 2008. Roelof Salters, Patrick van de Steeg, Jwalant Mishra, Dick Klaassen and Theo Beelen (all NXP Semiconductors) are acknowledged for many fruitful discussions. The current text contains minor corrections.

increasingly hard to guarantee correct SRAM operation under all process, voltage and temperature conditions. This translates to very tough requirements on SRAM parameters like Static Noise Margin (SNM), Write Margin (WM) and Read Current (I_{read}).

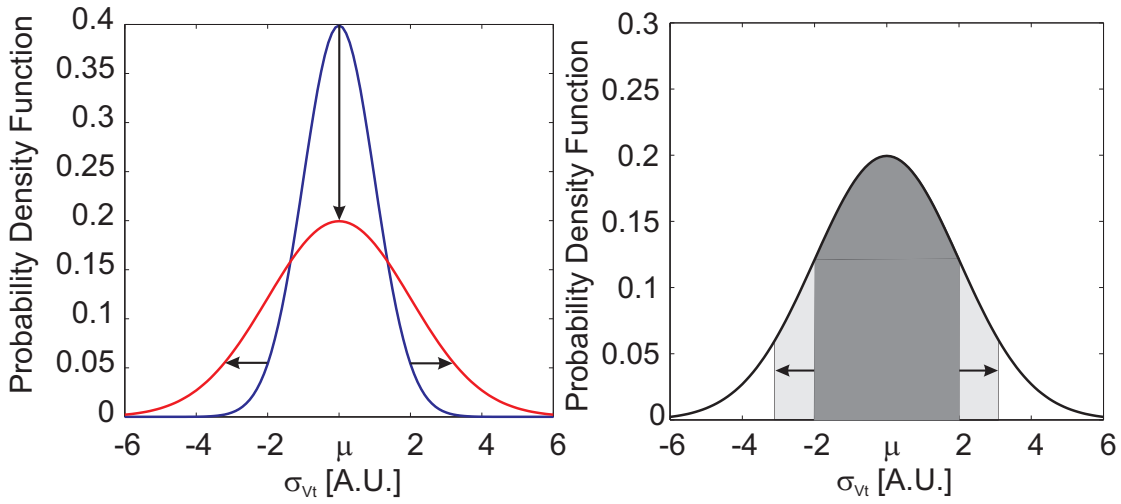

Figure 6.1: Increased variability leads to widening mismatch distributions (left). Increasing the number of memory bits per SoC leads to a larger part of the mismatch distribution being taken into account in memory bitcell design (right). [A.U.] stands for [Arbitrary Units]

.

SRAM yield should not be limited by parametric yield loss due to variability of design parameters. To guarantee no more than 0.1% yield loss for a 10Mb SRAM, a failure probability of $P_{\text{fail}} \leq 10^{-10}$ is taken into account in SRAM bitcell design for all relevant parameters. Provided the probability distribution is Gaussian, $P_{\text{fail}} \leq 10^{-10}$ corresponds to $\mu - 6.4\sigma$ (with μ the mean and σ the standard deviation of the distribution). Using Monte-Carlo (MC) simulations, the 6.4σ limits of the SRAM parameter distributions are estimated. Accurate estimation of the relevant parameters at $\mu - 6.4\sigma$ with plain Monte-Carlo takes billions of simulations and is too time consuming. Hence, a limited number of simulations is done $(10^3 - 10^4)$, the μ and σ of the distribution are extracted and $\mu - 6.4\sigma$ is determined by extrapolation. This technique is not always accurate, since the SNM distribution is not Gaussian at all [9, 47] and the distribution of I_{read} is not Gaussian in its tail.
This chapter presents the use of the simplest form of Importance Sampling (IS) to drastically increase the accuracy of Monte-Carlo simulations. This technique was applied before in a complex adaptive fashion, requiring complex sampling algorithms and post-processing [26]. This chapter presents a form of IS that requires less implementation effort. The applicability of the method is demonstrated by estimating the yield and probability distribution functions of SNM, WM and I_{read}. In the case of the SNM, a new method is presented for accurately estimating $P_{\text{fail}} \leq 10^{-10}$ by extrapolation. For SRAM Total Leakage Currents, it suffices to take the averages into account for designing SRAM cells and modules. A guideline is proposed to ensure Bitline Leakage Currents do not compromise SRAM functionality.

6.3 IMPORTANCE SAMPLING

Monte-Carlo analysis in circuit design normally assumes Gaussian distributed V_t-s of the transistors in the circuit. This results in many samples being drawn from around the average of the distribution. The extreme V_t-s are responsible for the extremes in the distributions of the output parameters (SNM, WM, $I_{\rm read}$, etc.). Therefore it makes sense to have more samples drawn from the tails of the V_t distributions. Using a Gaussian distribution with a larger standard deviation for the V_t is the simplest way to achieve this.

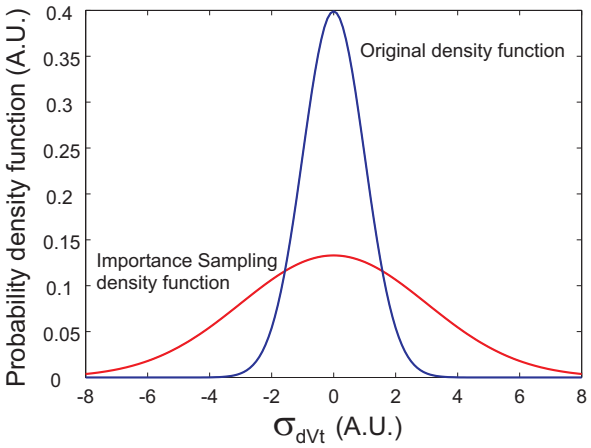

Figure 6.2: The principle of Importance Sampling. Using a density function with a larger standard deviation in Monte- Carlo analysis results in more samples being drawn from the extremes of the distribution. Here the latter density used a σ that was 3 times that of the original one.

From Figure 6.2 it is clear that using a wider Gaussian density function for Monte-Carlo sampling, indeed more samples are drawn from the extremes of the density. Using a wider V_t sampling distribution is a very practical choice, since no modifications to the circuit simulator are necessary. Using a wider density instead of the original distribution leads to distorted SNM, WM and $I_{\rm read}$ distributions. The correct density functions and distributions are obtained by a mathematical transformation based on the ratio of the original and IS distribution. The resulting distributions are now estimated over a much larger range compared to applying standard MC.
IS can be described more formally as follows. Suppose parameter x has a density $f(x)$. With IS, parameter x is sampled according to density $g(x)$. To compensate for sampling according to $g(x)$ instead of $f(x)$, the distribution function y, the sampled version of x, has to be multiplied by the ratio $f(x)/g(x)$. The sampled distribution function of parameter y is given by (6.1)-(6.2)

$$P(x<y) \;=\; F^{\rm IS}(y) \;=\; \frac{1}{N}\sum_{i=1}^{N} I_{\{x_i<y\}}\,\frac{f(x_i)}{g(x_i)}, \quad \text{with} \tag{6.1}$$

$$I_{\{x_i<y\}} \;=\; \begin{cases} 1 & \text{if } x_i < y \\ 0 & \text{if } x_i \geq y \end{cases}, \tag{6.2}$$

where N is the number of trials.

6.4 APPLICATION OF IS TO SRAM BIT CELL ANALYSIS

This section shows that with the same number of trials, IS can estimate much smaller failure probabilities than is possible with standard MC. It is also shown that extrapolated MC can lead to over- or under-estimation of the $P_{\text{fail}} \leq 10^{-10}$ for the most important SRAM parameters: SNM, I_{read} and WM. Moreover, for the SNM, a new method allows estimating $P_{\text{fail}} \leq 10^{-10}$ using extrapolated MC with high accuracy.
A 65nm SRAM cell is simulated using PSP MOS transistor models. A supply voltage $V_{dd} = 0.9V$ is used, to bring the cell closer to its operating limits. At this V_{dd}, the accuracy with which all parameters are determined becomes more important. The IS simulations use Gaussian distributions with a $\sigma = 3\sigma_{V_t}$ for the V_t-s of all transistors in the SRAM cell [we generated enough samples in the tails to draw conclusions].

6.4.1 STATIC NOISE MARGIN (SNM)

An SRAM cell has to be stable enough to be read without changing the data in the cell. The SNM is a measure for the read stability of the cell. The SNM is the amount of noise that can be imposed on the internal nodes of the SRAM cell before it changes its state. The SNM is determined by plotting the voltage transfer curve of one half of the SRAM cell together with the inverse of the voltage transfer curve of the other half of the cell. The sides of the largest squares that can be drawn inside the eyes are SNM_h ('high') and SNM_l ("low"), see Figure 6.3.

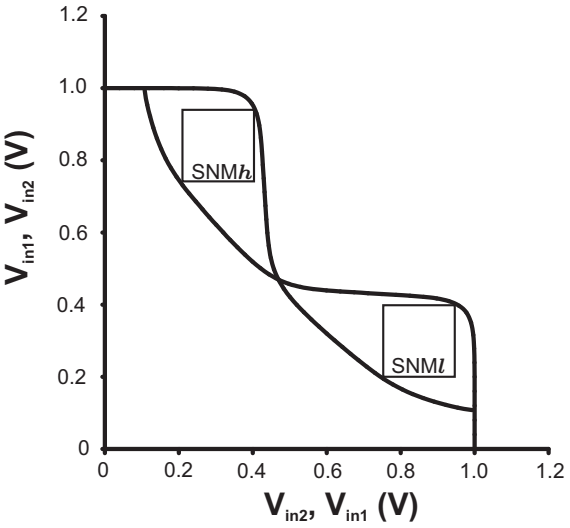

Figure 6.3: The butterfly curve of an SRAM cell, used to determine the SNM.

Both SNM_h and SNM_l have a Gaussian distribution. The minimum of SNM_h and SNM_l is traditionally defined as the SNM [47]. Since taking the minimum of SNM_h and SNM_l is a non-linear operation, the distribution of SNM is no longer Gaussian. Therefore using extrapolated MC to determine $P_{\text{fail}} \leq 10^{-10}$ does not yield accurate results.

Figure 6.4-left, shows the cumulative distribution function (CDF) of the SNM, determined by a MC simulation using 50k trials, both for standard MC (solid) and IS (dotted). Standard MC can

only simulate down to $P_{\text{fail}} \leq 10^{-5}$. Statistical noise becomes apparent below $P_{\text{fail}} \leq 10^{-4}$. Using the simple form of IS, $P_{\text{fail}} \leq 10^{-10}$ is easily simulated. The correspondence between Standard MC and IS is very good down to $P_{\text{fail}} \leq 10^{-5}$. Figure 6.4-left clearly shows that using extrapolated MC leads to overestimating the SNM at $P_{\text{fail}} = 10^{-10}$.

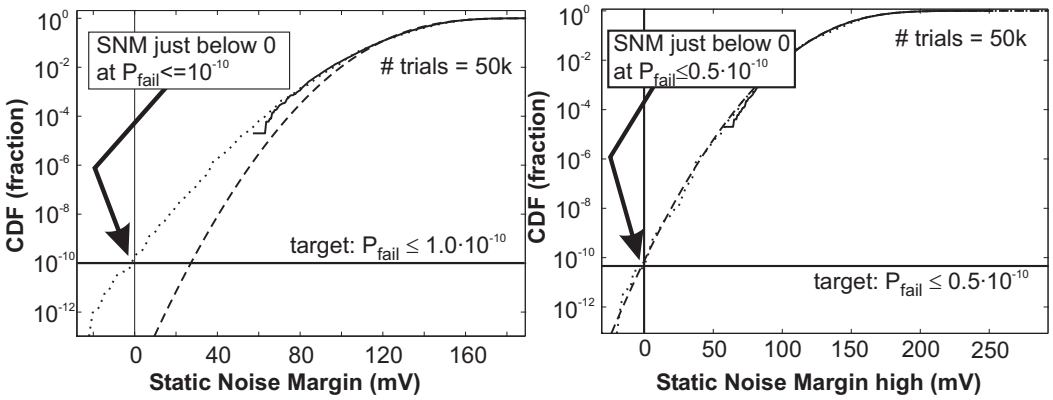

Figure 6.4: SNM (left) and SNM$_h$ ('high') (right) cumulative distribution function for extrapolated MC (dashed), standard MC (solid) and MC (dotted).

A new simple method is now presented to estimate the SNM by evaluating the distribution of only SNM$_h$ or SNM$_l$. Figure 6.4-right shows the CDF of SNM$_h$. The distribution of SNM$_h$ is a Gaussian distribution and extrapolation leads to a good estimate of SNM$_h$ at $P_{\text{fail}} = 10^{-10}$. The $P_{\text{fail}} = 10^{-10}$ limits for SNM$_h$ and SNM appear be to almost identical. At first sight, this is surprising, since the SNM and SNM$_h$ have different distributions. However, a small difference exists between SNM and SNM$_h$/SNM$_l$. The following describes how they are different.
The SNM is defined as the smaller value of SNM$_h$ and SNM$_l$

$$\text{SNM} \quad = \quad \min(\text{SNM}_h, \text{SNM}_l). \tag{6.3}$$

Next, we apply the probability rule

$$P(A \cup B) \quad = \quad P(A) + P(B) - P(A \cap B), \tag{6.4}$$

with $A = \{\text{SNM}_h \leq a\}$ and $B = \{\text{SNM}_l \leq a\}$. The probability that SNM$_h$ and SNM$_l$ simultaneously are very small is extremely low. Therefore, for the extreme chances, $P(A \cap B) \approx 0$ (and much smaller than the other chances). Assuming that SNM$_h$ and SNM$_l$ are identically distributed, it follows for the values of interest for a that:

$$\begin{aligned} P(\text{SNM} \leq a) &= P(\text{SNM}_h \leq a) + P(\text{SNM}_l \leq a) \\ &= 2P(\text{SNM}_h \leq a) \qquad (6.5) \\ &= 2P(\text{SNM}_l \leq a). \qquad (6.6) \end{aligned}$$

A failure probability for SNM$_h$ of $P(\text{SNM}_h \leq a) = 0.5\ 10^{-10}$ is required to get the same failure probability $P(\text{SNM} \leq a) = 10^{-10}$. In the example shown in this chapter, the difference between a for $P(\text{SNM}_h \leq a) = 0.5\ 10^{-10}$ and $P(\text{SNM} \leq a) = 10^{-10}$ is only 1.2mV, which

is within the statistical accuracy of IS. The justification is demonstrated in Figure 6.4 where the crossings are at the same for SNM. For larger values (and thus larger chances) the assumption that $P(A \cap B) = 0$ is not longer valid. Indeed Figure 6.4 shows that there (6.5) does not hold there (but this is also clear from the equation itself for large a).
The extrapolated version of $P(\mathrm{SNM}_h \leq a) = 0.5\ 10^{-10}$ deviates from $P(\mathrm{SNM} \leq a) = 10^{-10}$ by only 0.3mV. Effectively, using $P(\mathrm{SNM}_h \leq a) = 0.5\ 10^{-10}$ means extrapolating to $\sigma - 6.5\sigma$. This analysis shows it is possible to use extrapolated MC as an accurate estimate of the far tail of the SNM distribution.

6.4.2 READ CURRENT

The Read Current I_{read} is a measure for the speed of the memory cell and is therefore an important parameter. Figure 6.5 shows the extrapolated MC, regular MC and IS distribution for the Read Current of an SRAM cell. Again, there is a good match between regular MC (solid) and IS (dotted), down to $P_{\mathrm{fail}} \leq 10^{-4}$.

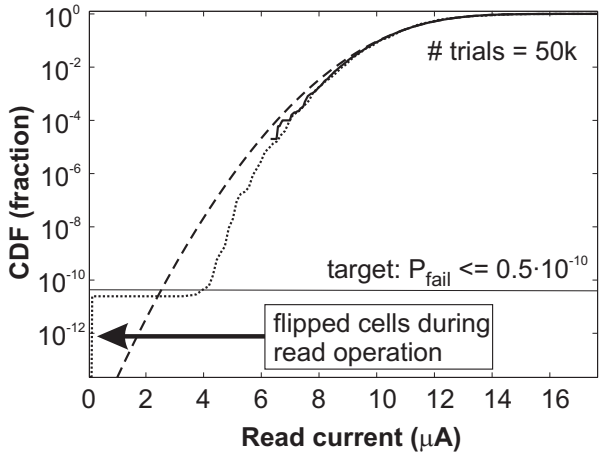

Figure 6.5: Read Current Cumulative Distribution function of the extrapolated distribution (dashed), regular Monte-Carlo (solid) and IS (dotted).

These Read Current simulations were done on one side of the cell. Therefore, $P_{\mathrm{fail}} \leq 0.5\ 10^{-10}$ has to be targeted for the Read Current as well. The correspondence with the SNM_h simulation is very good. The cells start flipping during a read action at almost exactly the same failure probability as where SNM=0mV.
These simulations show that extrapolated MC can result in serious underestimation of the Read Current. This can lead to over-design of the memory cell. To be able to accurately simulate the worst case Read Current as a result of mismatch, IS is essentially needed for sampling the Read Current I_{read} appropriately. Extrapolated MC is by no means accurate enough. This is in contrast to the SNM function.

6.4.3 WRITE MARGIN

An SRAM cell should not only be stable during read, it also has to be sufficiently instable to be written when desired. The Write Margin (WM) is a measure for the writeability of the SRAM

cell. A cell is written by precharging one bitline to $V_{\rm dd}$ and discharging the other bitline to ground, with the wordlines at $V_{\rm dd}$. The WM can be defined as the highest acceptable voltage on this low bitline (Figure 6.6).
For the WM, a similar line of reasoning holds as for the SNM. Therefore the target should be $P_{\rm fail} \leq 0.5\ 10^{-10}$. The distribution function of the WM was also simulated using extrapolated MC, standard MC and IS MC (Figure 6.7). Again, a good match is obtained between standard MC and IS MC. The WM is underestimated by about 10 mV, which is not a significant deviation. Therefore the far tail of the WM distribution can be estimated using extrapolated MC.

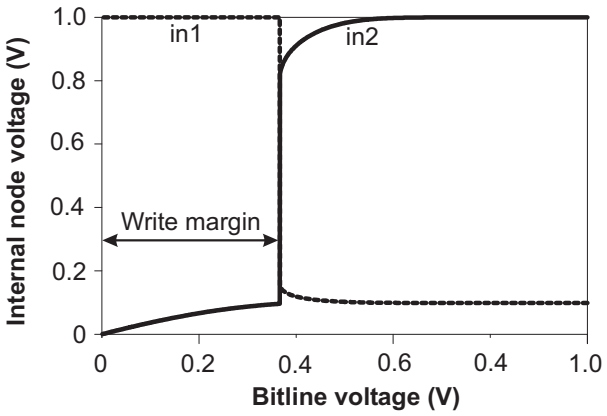

Figure 6.6: The internal node voltages of an SRAM cell versus the low bitline voltage. The write margin (WM) is defined as the highest bitline voltage at which de SRAM cell flips.

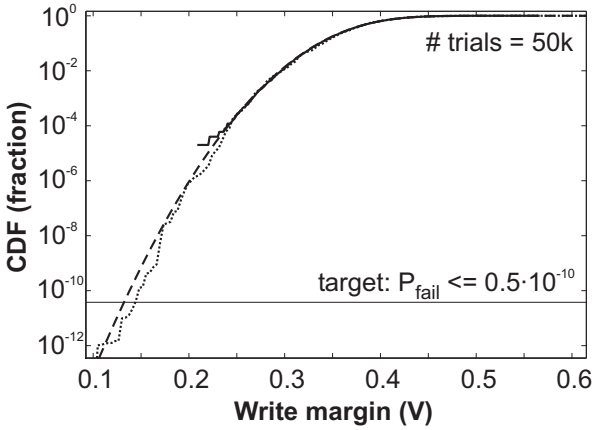

Figure 6.7: Write Margin (WM) Cumulative Distribution function of the extrapolated distribution (dashed), regular Monte-Carlo (solid) and IS Monte-Carlo (dotted).

6.4.4 LEAKAGE CURRENTS

Leakage Currents can be divided into two important categories:

- **Total Leakage Current:** Total Leakage Current is important for the standby power consumption of the memory. This can be estimated by multiplying the average of the total cell leakage by the number of cells in the memory instance. The large number of cells in an SRAM results in a small variation on this estimate, making this method sufficiently accurate.

- **Bitline Leakage Current:** Bitline Leakage is the sum of the leakage currents of the non-selected cells in the column being accessed. Too much Bitline Leakage Current can result in a non-functional memory. During reading, one of the two bitlines of the column is discharged to develop sufficient differential voltage for the sense amp to be detected. In a worst case situation, all non-accessed cells connected to the column being read are discharging the opposite bitline with their leakage currents. If the sum of the leakage currents is in the order of the worst-case Read Current, there is a risk of developing insufficient differential voltage on the bitlines and a read failure.
  Short columns with fewer cells have lower Bitline Leakage Currents than longer columns. Hence, if a memory with long columns can handle the worst case bitline leakage, a smaller instance of that memory with shorter columns can also handle the bitline leakage.

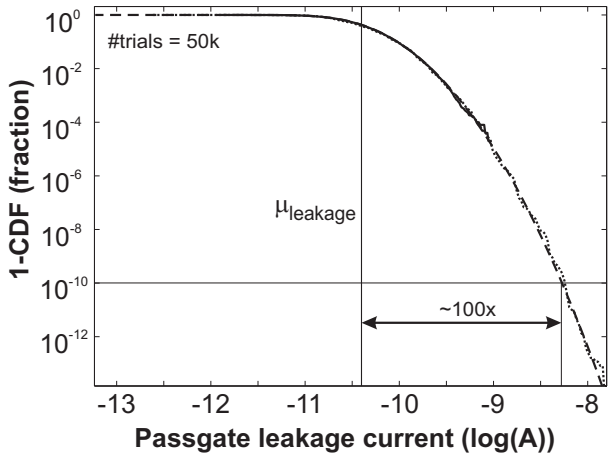

Figure 6.8: 1-CDF of the logarithm of the Bitline (Passgate) Leakage Current (leakage current of one cell): extrapolated MC (dashed), regular MC (solid) and IS MC(dotted).

Figure 6.8 shows the logarithm of the Bitline (Passgate) Leakage Current. Since the leakage current depends exponentially on the transistor treshold voltage V_t, the distribution of the logarithm is excellently Gaussian. The probability of a Bitline Leakage Current that is 100x higher than the average is approximately $P(I_{\mathrm{leak,bl}} \geq 100\ I_{\mathrm{leak,bl},\mu}) \approx 10^{-10}$ for this cell, meaning this is a very rare event. Hence it is safe to assume only one cell has worst case leakage and all other cells have an average leakage current. Inequality (6.7) is proposed as a guideline to ensure that Bitline Leakage Current does not compromise SRAM functionality

$$I_{\mathrm{read,wc}} \quad \geq \quad x\ (I_{\mathrm{leak,bl},6.4\sigma} + (L-2)\ I_{\mathrm{leak,bl},\mu}), \tag{6.7}$$

where $I_{\mathrm{read,wc}}$ is the worst case Read Current, L is the maximum number of cells in a column and x is a margin factor at the discretion of the designer.

6.5 CONCLUSION

Continuous scaling according to Moore's law and an increasing number of bits used in SRAM memories strongly increase the need for incorporating statistical information into the design of SRAM bit cells. To guarantee sufficient yield for a 10 Mb SRAM, failure probabilities of $P_{\text{fail}} \leq 10^{-10}$ are required, probabilities found in the far tails of the parameter distributions. Accurate statistical techniques are a must to be able to simulate such failure probabilities.
In this chapter it was shown that accurate statistical DC SRAM cell simulations are possible using a relatively simple statistical technique like Importance Sampling (IS) Monte Carlo (MC) with widened V_t distributions. The technique has been successfully applied to accurately estimate the distributions of Static Noise Margin (SNM), Write Margin (WM) and Read Current I_{read}.
For the SNM, it is shown that extrapolation of standard MC simulations overestimates the yield. In addition to the benefit of IS MC simulations, it has been shown that extrapolation of the Gaussian distributions of the individual eyes yields results in accurate yield estimation. The results of the latter method are in agreement with IS MC simulations.
The Read Current distribution deviates strongly from a Gaussian distribution and therefore its distribution can not be extrapolated. The use of extrapolated distributions would result in a pessimistic I_{read} and could thus lead to over-design of the memory cell and/or memory architecture. Importance Sampling or a technique with similar statistical accuracy is required to make correct decisions in the design process.
The WM can be estimated with extrapolated Gaussian distributions. Although a small difference of the WM at $P_{\text{fail}} \leq 10^{-10}$ is observed between extrapolated MC and IS MC, this difference is not significant.
To determine the SRAM Total Leakage Currents the average current per cell is multiple by the number of cells in the instance. A guideline is proposed to guarantee that Bitline Leakage Currents do not compromise SRAM functionality.

Section 7

Recommendations for PSTAR [42]

In this chapter we give some recommendations to allow for Importance Sampling.

- **Input:** The user should be able to define a distribution g for selecting parameters: by referring to a standard one (normal, uniform, lognormal, etc), or by explicitly defining a function. In the last case Pstar should perform some checks (like being positive, cumulative probability adds up to 1).
  The user should also be able to define the reference distribution f that would have been used without Importance Sampling.
- **Output:** Several items can be listed
  - Provide $P(X < t)$ for given output function X and given value of t.
  - Also allow for a PDF and a cumulative probability (CDF) plot: plot $P(X < t)$ for a list of t-values.
    This will need binning, hence an additional binning specification must be included. As default number of bins, the square root of the number of samples may be chosen. Creating a CDF by 'binning' can be improved by applying proper quadrature, like the Trapezoidal Rule.
  - In doing the current research it appeared to be very helpfull that plots of the values of f and of g at the sampling points can be plotted together.
  - A parameter sweep of the values of the quantities of interest at the chosen probability against the swept parameter(s) is wanted as well.
- **Generalizations:** Allow also for correlations. Include a functionality that determines g automatically (for instance by adaptivity). Also allow to determine t such that for given ε one has $P(X < t) < \varepsilon$

Section 8

Conclusions

A 0.1% yield loss for 10Mbit SRAM memory, which means that 1 in 10 billion cells fails ($P_{\text{fail}} \leq 10^{-10}$) can be efficiently estimated by Monte Carlo methods that are tuned by Importance Sampling. Importance sampling brings Monte Carlo to the area in parameter space from where the rare events are generated. By this a speed up of several orders can be achieved when compared to standard Monte Carlo methods. The efficiency of the method increases when the dimension of the parameter space increases.
The method would be a valuable extension to the statistical capacities of Pstar [42] and/or Spectre [49]. In fact the method can be efficiently implemented in any simulator and can be extended to allow for adaptive tuning of the rare event density distribution.
A version of Importance Sampling has been implemented using PStar with Matlab post processing and has been demonstrated to work correctly. The method has been applied to estimate the probability distribution of all 4 SRAM cell parameters: Static Noise Margin (SNM), Write Margin (WM), Read Current and Bitline Leakage Current. A good correspondence of Importance Sampling Monte Carlo and traditional Monte Carlo simulation was shown for the relevant probability range.
For the SNM, it is shown that extrapolation of standard MC simulations overestimates the yield. In addition to the benefit of ISMC simulations, it has been shown that extrapolation of the Gaussian distributions of the individual SNM eyes yields results in accurate yield estimation. The results of the latter method are in agreement with IS MC simulations.
The Read Current distribution deviates strongly from a Gaussian distribution and therefore its distribution can not be extrapolated. The use of extrapolated distributions would result in a pessimistic Read Current and could thus lead to over-design of the memory cell and/or memory architecture. Importance Sampling or a technique with similar statistical accuracy is required to make correct decisions in the design process.
The WM can be estimated with extrapolated Gaussian distributions. Although a small difference of the WM at $P_{\text{fail}} = 10^{-10}$ is observed between extrapolated MC and IS MC, this difference is not significant.
To determine the SRAM Total Leakage Currents the average current per cell is multiple by the number of cell in the instance. A guideline is proposed to guarantee that Bitline Leakage Currents do not compromise SRAM functionality.

We introduced Importance Sampling as a technique to efficiently perform failure analysis. To prove benefits over standard Monte Carlo we applied and extended knowledge from Large Deviation theory. The basics of the method can easily be implemented in a circuit simulator or in a

shell procedure around a circuit simulator. For a refined procedure, involving adaptive sampling, we introduced a new approach. Here some intial tests were made using 1-dimensional functions. The real benefit must come from problems with parameters in a higher dimensional space. This will require further research.
Apart from the studied Importance Sampling we also described two additional variants (weighted importance sampling, regression importance sampling) that have some benefits from a numerical mathematics point of view, but for us the obtained benefits are here not decisive.
In further improving the performance of a particular variant of Importance Sampling the variance can be minimized by optimizing a parameter. Apart from some trivial situations, in general this requires some accurate numerical procedures.

RESPONSE SURFACE MODELING

We did some minor experiments with Response Surface Modeling (RSM) techniques (using the MatLab M3/SUMO toolbox developed at the University of Antwerp/Ghent[1]). After paying the costs for exploring the design space and to build the model, the output function enables a rapid Monte-Carlo simulation. To give an impression, for 4 SRAM functions (SNM, WM, I_{Read}, I_{Leakage}) and 6 V_T-s, 10 million samples can efficiently be simulated within ca 7 minutes, which is a speed-up of 1400 with respect to Pstar [42] (that used 1000 trials per minute). For standard MC using 10^{10} samples, Pstar will need more than 10^7 minutes ($= 1.7\,10^5$ hours, or $\frac{2}{3}\,10^4$ days, or ca 20 years), while MC via RSM was done in 7000 minutes. The cost for generating the RSM was 10h.
For RSM techniques one needs to verify the accuracy of the models (sometimes strange peaks occur in the surface). Also the model has to become very accurate in the area that is important for the failure analysis.

In the tool ROAD (RObust Analog Design) of ExtremeDA (http://extreme-da.com/ROAD_-Suite.html) a Quadratic Response Surface Model (QRSF) for a nonlinear performance function f is constructed [32, 33, 34]. The main features are

- One efficiently determines high-order moments of QRSF via a "binomial moment evaluation".
- Next one determines the poles b_i and residues a_i of a transfer function $H(s) = \sum_{i=1}^{M} \frac{a_i}{s-b_i}$ such that moments in t-domain match those of QRSF.
- The pdf(f) is obtained via a time impulse response $h(t) = \sum_{i=1}^{M} a_i e^{b_i t}$ (actually $t \approx f$)
- The cdf(f) follows simply via $s(t) = \int_0^t h(\tau) d\tau$.
- One applies a carefull shifting: pdf($f \pm f_0$).

Here the step via the moments is the unattractive part, despite the nice recursion. The problem with exlicit moment matching algorithms always is the stability of these recursions. However the approach is interesting when viewed from the point of view of Model Order Reduction (MOR) where poles and residues are calculated using other techniques. Some research is needed here to obtain the required implicit moment matching.

[1] http://www.sumo.intec.ugent.be/

FUTURE WORK

Topics to be studied further are listed below.

- The outcome of an evaluation can guide to how to determine specific sampling points. Our current experiments did not exploit this as well. Section 4 provides a starting procedure to adaptively sample points at proper locations. By this one approximates the optimum Importance Sampling function g in an adaptive way.

- The sampling function g may be different for various output responses. Currently we have derived a practical form of Importance Sampling for SRAM cell simulations. However, we have not solved the general question: What is the optimal Importance Sampling distribution g for statistical (SRAM) circuit simulations? Again, Section 4 provides a starting procedure for this.

- How many trials are needed to obtain the required accuracy? In Section 2.4 we derived by the Large Deviation principle that one may need a number $N \approx 1/p$ for Monte Carlo. In Section 3.2 we proved that Importance Sampling needs less samples to obtain a reduction of the variance of the estimate of p. Both proofs were not trivial. In practice we worked with much less samples (order $\mathcal{O}(\frac{1}{\sqrt{p}})$). One needs some adaptive error estimates during the sampling process.

- For the SRAM response functions the distributions have been determined. For low voltage memory quite a number of parameters have influence. Sensitivity has not yet been exploited in our experiments. A simple procedure that applies when sensitivity analysis is not provided by a circuit simulator is to run a MC "scan" in advance putting variability on all parameters at once. Now, before calculations of the distributions, dominant parameters can be determined (note that parameters can be dominant in a specific region only). Only these parameters need to be included in the calculation of the distribution. To exploit sensitivity in MC see [19].

- Which part of the input parameter space meets the output specifications of the SRAM simulations: *i*.e., for given ε find t such that $P(X < t) \leq \varepsilon$. How to detect the part of the input parameter space that determines output specifications for SRAM simulations?
  This relates to Inverse Problem techniques [25, 30]. Note that we have learned during the research that analog designers are interested in deriving a Cumulative Probability Function. It means that one is interested in a sequence of ε's.

- Which methods other than Importance Sampling can be used to improve accuracy and performance (to increase speed) of statistical runs for SRAM. How can they be applied, or combined with Importance Sampling. For example, how can Latin Hypercube Sampling or generalized Polynomial Chaos (gPC) Theory [2, 37, 54] be combined with Importance Sampling (see for instance [39])? Perhaps that on the short term these questions are more interesting to be answered than to generalize adaptive importance sampling.

- How can Response Surface Modelling Techniques be used to further reduce evaluation time, *e.g.*, by determining the dominant parameters [28, 52]. We note that [32, 33, 34] have described a procedure to efficiently obtain statistical moments for nonlinear response functions based on approximation techniques from Model Order Reduction.

Appendix A

Matlab Code

A.1 File Matlab_ImpSampling_Pstar.m

```
clear;

%path = '/home/nlv15606/projects/C065/lop/sram_opt/pstar/sram/';
path = 'H:\__RESEARCH_PAPERS__\Statistics\SRAM\Importance_Sampling\';

% Plotting chances y=10^(-k) vs x; y=N(x) the cumulative normal density function
% y=N(x)=0.5 ( 1+erf(x/sqrt(2) ), hence x=sqrt(2) erfinv(2y-1)

% Powers: -12, -11-0.75, -11-0.5, -11-0.25, -11, ..., -1, -0.75, -0.5, -0.25, 0
powers = linspace(-12,0,49)
y=10.^powers
x=sqrt(2) * erfinv(2*y-1)

figure(1);
subplot(121), h1=semilogy(x,y,'b');
title('{\bf Log(Cumul. Normal chances)}');
grid minor;

subplot(122), h2=plot(x,y,'b');
title('{\bf Cumul. Normal chances}');
grid minor;

[nx,mx]=size(x);
[ny,my]=size(y);

xpos= - x([mx-2:-1:1]); % Mirror only the negative values of x
ypos= 1 - y([my-2:-1:1]);

xtot=[x(1,1:mx-2) xpos(1,1:mx-2)];
ytot=[y(1,1:my-2) ypos(1,1:my-2)];

figure(2);
subplot(121), h1=semilogy(xtot,ytot,'b');
title('{\bf Log(Cumul. Normal chances)}');
grid minor
subplot(122), h2=plot(xtot,ytot,'b');
title('{\bf Cumul. Normal chances}');
grid minor

% Data files from Pstar: 8 columns, containing "index, vt1--vt6, snm"
% It is assumed that the vtj are mutually independent
%
% snm=snm(vt1,vt2, ..., vt6) is output result from Pstar
%
```

```
w=waitbar(0,'Reading data files ...');
waitbar(0,w);
file_uni  = 'snm_1e5_uni.table';
data_uni  = single(load([path file_uni])); % uniformly distributed [mu-6sigma, mu+6sigma]
waitbar(0.5,w);

file_norm = 'snm_1e5_norm.table';
data_norm = single(load([path file_norm])); % normally distributed with stdv=sigma

waitbar(1,w);

close(w);

pstarSigma = 6;
numsam = length(data_uni); % The number of parameter tuples (vt1,vt2, ..., vt6)
%numsam=100000;
numbin = 250;

[mdu,ndu] = size (data_uni); % n=8 in example
vt  = data_uni(:, [2:ndu-1]);
snm = data_uni(:, ndu);

snm_norm = data_norm(:,ndu);

figure(3);

[n_snm,snm_bin_centers]=hist(snm,numbin);
[n_snm_norm, snm_norm_bin_centers]=hist(snm_norm,numbin);

subplot(121), bar(snm_bin_centers,n_snm); % Fig 5.2a
title('{\bf Histogram SNM with V^T_k uniformly distributed on [\mu-6\sigma, \mu+6\sigma]}','FontSize',12)

subplot(122), bar(snm_norm_bin_centers,n_snm_norm); % Fig. 5.2b
title('{\bf Histogram SNM with V^T_k normally distributed with N(\mu,\sigma)}','FontSize',12);

w=waitbar(0,'Calculating correlations of vt-s...');
waitbar(0,w);
'Correlation data of vt-s:'
[rho_uni,pval_uni]=corr(vt) % rho_uni contains positive and negative values;
                            % pval_uni is nonnegative
rho=rho_uni;
for i=1:ndu-2
  rho(i,i)=0;
end
rho_max=max(max(rho));
rho_min=min(min(rho));

figure(4);
xc=[1:1:ndu-2];
yc=xc;
[XC,YC]=meshgrid(xc,yc);

plot3(XC,YC,rho);
axis([1 ndu-2 1 ndu-2 rho_min rho_max]);
title('{\bf Corr between vt-s; diag (was 1) set to 0}');
grid;

waitbar(1,w);
close(w);

w=waitbar(0,'Starting calculating probabilities ...');
waitbar(0,w);

%
% Importance Sampling using samples from a broad uniform distribution (data_uni)
%

snmRange = max(snm)-min(snm)
```



```
vtmean = mean(vt)

% The sigmaVt of the normal distribution can be calculated from
% sigmaVt of the uniform distribution.
sigmaVt_uni = std(vt)
sigmaVt_uni_scaled = sigmaVt_uni/pstarSigma
sigmaVt_norm = sigmaVt_uni_scaled*sqrt(3)

% The sigmaVt of the normal distribution can also be calculated
% from the range of the uniform Vt distribution.
vtstd = (max(vt)-mean(vt))/pstarSigma

% From PStar we obtain the next numbers for sigmaVt (trials=10000):
%vtstd = [0.0396 0.0350 0.0389 0.0396 0.0350 0.0389]

% The exponent of the normal distribution -0.5*((vt - u)/s)^2
%
[mvt,nvt]   = size(vt);
vtmeanM = ones(mvt,nvt)*diag(vtmean);
expMult = ( (vt-vtmeanM)*diag(1./vtstd) ).^2 ;
f_pdf   = 1/sqrt(2*pi) * exp( - 1/2 * expMult ) * diag(1./vtstd) ;

% We have to determine the chance p(snm<X) for several values of X
% This is done by the formula [using the mutual indpendency]
%
% p(snm<X) = (1/N) sum InX(snm) * f_pdf(vt1)/g_pdf(vt1) * ... * f_pdf(vt6)/g_pdf(vt6)
%
% Here InX(x) = 1 if x<=X, InX(x) = 0 if x>X.
%
% g_pdf is from a uniform distribution between min(vt) and max(vt): 1/vtRange
%
% f_pdfMult = f_pdf(vt1)* ... * f_pdf(vt6)
f_pdfMult = prod(f_pdf,2);
vtRange = max(vt)-min(vt) ;
% 1/N * 1/g_pdfMult = 1/N * prod(vtRange)
one_over_N_times_one_over_g_pdfMult = prod(vtRange)/numsam ;
% 1/N * 1/g_pdfMult = 1/N * prod(vtRange)
% 1/n * prod_k (f_k/g_k)
pdfMultNorm = f_pdfMult * one_over_N_times_one_over_g_pdfMult;

waitbar(0.5,w);
[snmAxis, snmPDF, snmCDF] = Matlab_Makepdf(snm,pdfMultNorm,numbin);
waitbar(1,w);


% ====================== Normal/Standard Monte-Carlo ======================
% (via Histogram sampling, covering non-normality of output function)
%

% normHist contains totals in each bin; normbins: bin-centers
[normHist,normBins] = hist(snm_norm,numbin);
snmRangeNorm = max(snm_norm)-min(snm_norm);
stepsizeNorm = snmRangeNorm/numbin;
snmPDFnorm = normHist'/numsam; % Calculate fraction in each bin interval
snmCDFnorm = cumsum(snmPDFnorm); % Sum all chances
snmAxisNorm = normBins; %

% ================ Extrapolated Monte-Carlo ================
% (via Histogram sampling, assuming normal density function of output function)
%

meanNorm = mean(snm_norm) ;
sigmaNorm = std(snm_norm) ;

snmPDFextr = 1/sqrt(2*pi)/sigmaNorm*exp(-1/2*(snmAxisNorm-meanNorm).^2/sigmaNorm^2);
snmCDFextr = cumsum(snmPDFextr*snmRangeNorm/(numbin-1));
```

```
figure(5)  % Fig. 5.1
h3=semilogy(snmAxis',snmCDF,'b', snmAxisNorm,snmCDFnorm,'g--',  snmAxisNorm, ...
            snmCDFextr,'r-.');

set(h3,'LineWidth',2);

legend('{\bf Importance Sampling MC}', '{\bf Normal MC}', '{\bf Extrap. MC}', ...
       'Location', 'SouthEast');
title('{\bf Comparison of Cumulative Probabilities}','FontSize',12);
xlabel('{\bf X}','FontSize',12,'FontAngle','italic');
ylabel('{\bf P(x<X)}', 'Rotation', 0, 'FontSize',12,'FontAngle','italic');
text(0.055, 10^3, '{\bf x=SNM(V^T_1,...,V^T_6)}','FontSize',12);
grid;

hold on;

%msu=mean(snm);
%msn=meanNorm; %mean(snm_norm);
%semilogy([msu,msu],[10^5,10^(-20)],'m-..');
%semilogy([msn,msn],[10^5,10^(-20)],'m-..');
%text(0.055,10^1,strcat(strcat('{\bf \mu_{unif}(x)=',num2str(msu)),'}'));
%text(0.055,10^(-1),strcat(strcat('{\bf \mu_{norm}(x)=',num2str(msn)),'}'));

close(w);
```

A.2 File Matlab_Makepdf

```
%
% [axisbin, pdfbin, cdfbin] = Matlab_Makepdf(values, probvalues, nobins)
%
% Creates the pdf, cdf (with 'nobins' bins) and accompanying axis
% for random data consisting of 'values', based on the probabilities
% of those values.
%

function [axisbin, pdfbin, cdfbin] = Matlab_Makepdf(values, probvalues, nobins)

axisbin = zeros(nobins+1,1);
pdfbin  = zeros(nobins+1,1);
cdfbin  = zeros(nobins+1,1);

minValue = min(values);
maxValue = max(values);
rangeValue = maxValue-minValue;
binstep = rangeValue/nobins ; % binlength
axisbin(:,1) = [minValue:binstep:maxValue]'; % Always one point more than the number of bins
[na,ma] = size(axisbin);
[nv,mv] = size(values);

[sortedValues,sortingIndex] = sort(values(:,1));
[nsv,msv] = size(sortedValues);
[nsi,msi] = size(sortingIndex);
[np,mp]   = size(probvalues);

kstart=0;
for jbinbound=1:1:na
   bool=0;
   k=kstart;
     % occurrences in (X,X+step]
     while ( (k < nv) & (sortedValues(k+1,1) <= axisbin(jbinbound,1)) )
        k=k+1;
        bool=1;
     end
     if (bool)
```



```
        % sum all the probabilities of the occurrences
        pdfbin(jbinbound,1) = sum(probvalues( sortingIndex(kstart+1:k,1), 1));
        kstart=k;
     else
        kstart=k+1;
     end
end
cdfbin = cumsum(pdfbin);
```

Appendix B

Source Code Adaptive Importance Sampling Simulation

For the simulations in the 1-d testbed in Chapter 4, we used the following routines programmed in the open source statistical programming language R (see www.r-project.org). Defining a function f consisting of a combination of the three basic 1-d-phantoms, we can explore the dependence of the algorithm on the three relevant control parameters.

```
#########################################################
#
#       F I R S T  E X P L O R A T I O N
#
#
#
#
#
#########################################################


#########################################################
#
#      D E F I N E  T H E  F U N C T I O N  T O  B E
#
#             E X P L O R E D
#
#########################################################


# f <- function(t){x <- 1/(t-5)^2}     # spike phantom
# f <- function(t){x <- 0.5*t^2}       # outback phantom
# f <- function(t ){x <- 100*(1-100*(t-4)^2)}   # bump phantom

f <- function(t){x <- 1/(t+4)^2 + 0.5*t^2 + 100*(1-100*(t-4)^2)}

#########################################################



#########################################################
#
#    E X P L O R A T I O N  P A R A M E T E R S
#
#########################################################


# exploring the function f ###################################
```



```
alpha <- 5

# exploration width ########################################

width <- 2

# exploration gain #########################################

gain <- 200

# initiating exploration state ###################################

points <- c()

weights <- c()

numbers <- 1

#########################################################

# macro sampling step ######################################

sample <- function()
{
n <- 0
x <- rnorm(1)

while(f(x) < alpha)
{
x <- rnorm(1)
}
}

#########################################################
#
#        E X P L O R A T I O N
#
#########################################################

# 1 first exploration step ###################################

points[1]<- sample()
weights[1] <- 1
stepnumber <- 1


# 3. loop with stopping criterion ##############################

while(stepnumber < gain * numbers)
{
x <- sample()
stepnumber <- stepnumber + 1

# 2 check distance/outback and modify state ###################

dist <- abs(points - rep(x,numbers))
d <- min(dist)
outback <- 8.5

if(d > width)                    # 2a
{
if(abs(x) < outback)
{
numbers <- numbers + 1
points[numbers] <- x
weights[numbers] <- 1
}
```

```
else
{
k <- which.min(dist)
weights[k] <- weights[k] + 1
}
}
else                              # 2b
{
if(abs(x) < outback)
{
k <- which.min(dist)
weights[k] <- weights[k] + 1

if(f(points[k]) < f(x))           # accept
{points[k] <- x}
else                              # reject
{}
}
else
{
k <- which.min(dist)
weights[k] <- weights[k] + 1
}
}

#########################################################

}

# end loop ###############################################

status <- c(points,weights)

print(status)                     # output

#########################################################
#
#       E N D  F I R S T  E X P L O R A T I O N
#
#########################################################


#########################################################
#
#       E X P L O R A T I O N  F R O M  S E E D
#
#########################################################

EFS <- function(status, alpha, gain)   % gain is the stopping factor E
{

L <- length(status)*0.5
newstatus <- c()
newpoints <- c()
newweights <- c()
n <- 1

for(i in 1:L)
{
w <- 0
r <- L + i
M <- status[r]*gain

for(j in 1:M)
    {
    val <- status[i] + rnorm(1,0,0.5)
    if(f(val) > alpha)
        {
```



```
        w <- w + 1
        if(f(val) > f(status[i]))
            {
            status[i] <- val
            }
            else{}
        }
        else{}
    }

if(w > 0)
    {
    newpoints[n] <- status[i]
    newweights[n] <- status[r] + w
    n <- n+1
    }
    else{
        if(f(status[i]) > alpha)
            {
            newpoints[n] <- status[i]
            newweights[n] <- status[r]
            n <- n + 1
            }
            else{}
        }
}

newstatus <- c(newpoints,newweights)
print(newstatus)
}


#########################################################
#
#       E N D  O F  E X F R O M S E E D
#
#########################################################



#########################################################
#
#        W E I G H T E D  S A M P L I N G
#           with 'sweeps' number of simulation steps
#
#########################################################

WS <- function(points,weights,alpha,sweeps)
{
## mixture probabilities ###################################

NN <- rep(stepnumber, length(weights))
prob <- weights/NN

mixture <- rmultinom(sweeps,length(weights),prob)

z <- c()
approx <- 0

for(j in 1:length(weights))
{
z[j] <- 0
for(i in 1:sweeps){z[j] <- z[j] + mixture[j,i]} # samples from jth variable

mixfrom <- rnorm(z[j],points[j])
for(u in 1:z[j])
{
if(f(mixfrom[u]) > alpha)
```

```
{approx <- approx + exp(points[j]^2/2 - mixfrom[u]*points[j])}
else
{}
}

}

## approximation ########################################

print(approx / sum(mixture))
}
#########################################################
#
#       E N D  W E I G H T E D  S A M P L I N G
#
#########################################################
```


Appendix C

Alternatives For Histograms

The shape of a distribution (unimodality, asymmetries etc.) is difficult to assess from a normal probability plot. For this we need to estimate the density and present it in a plot. A widely used density estimator (although it is not always recognized as such) is the histogram. Let $X_1, \ldots, X_n$ be a random sample from a distribution function F (pertaining to a law P) on $\mathbb{R}$, with continuous derivative $F' = f$. As before, we denote the empirical distribution function by P_n. Let I be a compact interval on $\mathbb{R}$ and suppose that the intervals $I_1, \ldots, I_k$ form a partition of I, i.e.

$$I = I_1 \cup \ldots \cup I_k, \quad I_i \cap I_j = \emptyset \text{ if } i \neq j.$$

The histogram of $X_1, \ldots, X_n$ with respect to the partition $I_1, \ldots, I_k$ is defined as

$$H_n(x) := \sum_{j=1}^{k} \frac{P_n(I_j)\, I_{I_j}(x)}{|I_j|},$$

where $|I_j|$ denotes the length of the interval I_j. It is clear that the histogram is a stepwise constant function. Two major disadvantages of the histogram are

- the stepwise constant nature of the histogram
- the fact that the histogram heavily depends on the choice of the partition

In order to illustrate the last point, consider Figure C where the two histograms are made from the same data set.
It is because of this phenomenon that histograms are not to be recommended. A natural way to improve on histograms is to get rid of the fixed partition by putting an interval around each point. If $h > 0$ is fixed, then

$$\widehat{N}_n(x) := \frac{P_n((x-h, x+h))}{2h} \tag{C.1}$$

is called the *naive density estimator* and was introduced in 1951 by Fix and Hodges in an unpublished report (reprinted in [16]) dealing with discriminant analysis. The motivation for the naive estimator is that

$$P(x-h < X < x+h) = \int_{x-h}^{x+h} f(t)\, dt \approx 2\, h\, f(x). \tag{C.2}$$

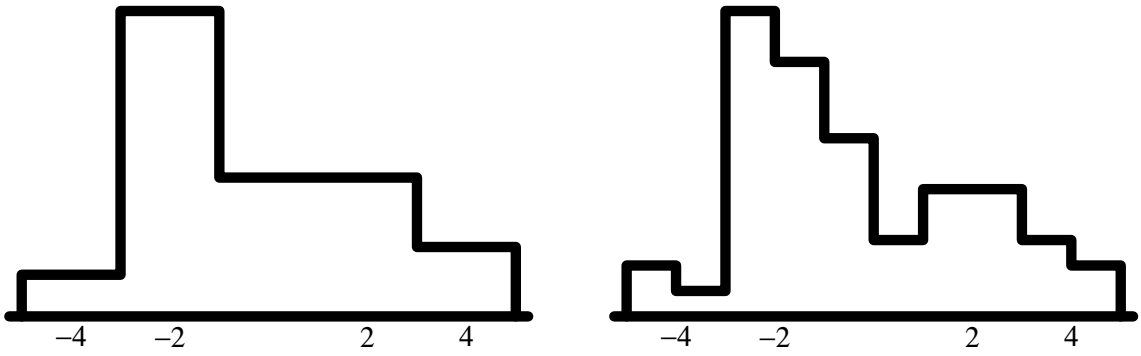

Figure C.1: Two histograms of the same sample of size 50 from a mixture of 2 normal distributions.

Note that the naive estimator is a local procedure; it uses only the observations close to the point at which one wants to estimate the unknown density. Compare this with the empirical distribution function, which uses all observations to the right of the point at which one is estimating.

It is intuitively clear from (C.2) that the bias of $\widehat{N}_n$ decreases as h tends to 0. However, if h tends to 0, then one is using less and less observations, and hence the variance of $\widehat{N}_n$ increases. This phenomenon occurs often in density estimation. The optimal value of h is a compromise between the bias and the variance. We will return to this topic of great practical importance when we discuss the MSE.

The naive estimator is a special case of the following class of density estimators. Let K be a *kernel function*, that is a nonnegative function such that

$$\int_{-\infty}^{\infty} K(x)\,dx = 1. \tag{C.3}$$

The *kernel estimator* with kernel K and bandwidth h is defined by

$$\widehat{f}_n(x) := \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - X_i}{h}\right). \tag{C.4}$$

Thus, the kernel indicates the weight that each observation receives in estimating the unknown density. It is easy to verify that kernel estimators are densities and that the naive estimator is a kernel estimator with kernel

$$K(x) = \begin{cases} \frac{1}{2} & \text{if } |x| < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Examples of other kernels are given in Table C.1. Kernel density estimators are available in the Statistics Toolbox of MATLAB through the command `ksdensity`, including an automatic choice of the bandwidth h. The default kernel is the Gaussian kernel (called normal in MATLAB), other available kernels are box, Epanechnikov and the triangular (called triangle in MATLAB) kernels.

name	function
Gaussian	$\frac{1}{\sqrt{2\,\pi}}\,e^{-\frac{1}{2}\,x^2}$
naive/rectangular	$\frac{1}{2}\,1_{(-1,1)}(x)$
triangular	$(1-\lvert x\rvert)\,1_{(-1,1)}(x)$
biweight	$\frac{15}{16}\,(1-x^2)^2\,1_{(-1,1)}(x)$
Epanechnikov	$\frac{3}{4}\,(1-x^2)\,1_{(-1,1)}(x)$

Table C.1: Well-known kernels for density estimators.

Appendix D

Discrete Probability Distributions

This chapter contains an overview of common discrete distributions, in alphabetical order. For more information on these distributions, we refer to [22]. Some generating functions can be expressed in terms of hypergeometric functions. For more information on these particular functions, we also refer to [22]. Capital X always refers to a random variable with the distribution being discussed.

Bernoulli distribution

A special case of the binomial distribution, namely $n = 1$. Often q stands for $1 - p$.

- Parameter: $0 \leq p \leq 1$
- Values: $0, 1$
- Probability mass function: $P(X = 1) = p$, $P(X = 0) = 1 - p$
- Expected value: p
- Variance: $p\,(1 - p)$
- Probability generating function: $p\,t + (1 - p)$
- Moment generating function: $p\,e^t + (1 - p)$.

Binomial distribution

The binomial distribution describes the number of successes among n independent trials with equal success probability p. Often q denotes $1 - p$. The binomial distribution is a special case of the multinomial distribution, with $m = 2$. The binomial distribution converges (in distribution) for $n \to \infty$ and $np = \lambda$ fixed to a Poisson distribution with parameter λ. For $p \leq 0.10$, the binomial distribution can be approximated by a Poisson distribution. For $np > 5$ *and* $n(1 - p) > 5$, the binomial distribution can be approximated by a normal distribution.

- Parameters: $n = 1, 2, \ldots$, $0 \leq p \leq 1$
- Values: $0, 1, \ldots, n$
- Probability mass function: $P(X = k) = \binom{n}{k} p^k\,(1 - p)^{n-k}$

- Expected value: np
- Variance: $np(1-p)$
- Probability generating function: $(p\,t+1-p)^n$
- Moment generating function: $\left(p\,e^t+1-p\right)^n$

Geometric distribution

This is a special case of the negative binomial distribution, with $r = 1$. The geometric distribution measures the number of independent trials, each with success probability p, until the first success (successful trial included in the total number). The geometric distribution has no memory, *i.e.*, $P(X > n+m \mid X > n) = P(X > m)$. It is the only discrete distribution with this property and is therefore the discrete counterpart of the exponential distribution.

- Parameter: $0 \leq p \leq 1$
- Values: $1, 2, \ldots$
- Probability mass function: $P(X=k) = p\,(1-p)^{k-1}$
- Expected value: $\dfrac{1}{p}$
- Variance: $\dfrac{1-p}{p^2}$
- Probability generating function: $\dfrac{p\,t}{1-(1-p)\,t}$
- Moment generating function: $\dfrac{p\,e^t}{1-(1-p)\,e^t}$

Hypergeometric distribution

The hypergeometric distribution counts the number of successes when n elements are selected *without replacement* from a group of N elements of which M mean "success" and $N-M$ imply "failure".

- Parameters: $N = 1, 2, \ldots$, $n = 0, 1, 2, \ldots, N$, $M = 0, 1, 2, \ldots, N$.
- Values: $\max(0, n-(N-M)), \ldots, \min(n, M)$
- Probability mass function: $P(X=k) = \dfrac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}$
- Expected value: $\dfrac{nM}{N}$
- Variance: $\dfrac{n\,M\,(N-M)\,(N-n)}{N^2\,(N-1)}$

- Probability generating function: ${}_2F_1[-n,-M,-N;1-t]$ where ${}_2F_1$ is a hypergeometric function.
- Moment generating function: ${}_2F_1[-n,-M,-N;1-e^t]$ where ${}_2F_1$ is a hypergeometric function.

Multinomial distribution

The multinomial distribution generalises the binomial distribution. Whereas a binomial distribution describes a sequence of independent Bernoulli experiments with each two possible outcomes (success and failure), the multinomial distribution describes a sequence of n mutually independent experiments with a fixed finite number m ($m \geq 2$) of possible outcomes. Let X_i denote the number of occurrences of the ith possible result ($i = 1, \ldots, m$) and p_i the probability that the ith possible result occurs in one experiment.

- Parameters: $n = 1, 2, \ldots$, $m = 1, 2, \ldots$, $0 \leq p_i \leq 1$ with $p_1 + \ldots + p_m = 1$
- Values: $\{(k_1, \ldots, k_m) \mid k_i \in \{0, 1, \ldots, n\}\ (i = 1, \ldots, m)$ and $\sum_{i=1}^{m} k_i = n\}$
- Probability mass function: $P\left((X_1, \ldots, X_m) = (k_1, \ldots, k_m)\right) = \dfrac{n!}{k_1! \ldots k_m!}\, p_1^{k_1} \ldots p_m^{k_m}$
- Vector of expected values: $(np_1, \ldots, np_m)$
- Covariance matrix: $\mathrm{Cov}(X_i, X_j) = -np_ip_j$ ($i \neq j$), $\mathrm{Var}(X_i) = np_i\,(1 - p_i)$
- Probability generating function: $\left(\sum_{i=1}^{m} p_i\, t_i\right)^n$
- Moment generating function: $\left(\sum_{i=1}^{m} p_i\, e^{t_i}\right)^n$

Negative binomial distribution

This distribution counts the total number of independent Bernoulli experiments with equal success probability p that is necessary to arrive at r successful experiments (the total number including the rth success). If U_i ($i = 1, \ldots r$) are mutually independent and all geometrically distributed with parameter p, then $X = \sum_{i=1}^{r} U_i$ has the negative binomial distribution with parameters p and r.

- Parameters: $0 \leq p \leq 1$, $r = 1, 2, \ldots$
- Values: $r, r + 1, \ldots$
- Probability mass function: $P(X = k) = \dbinom{k-1}{r-1}\, p^r\, (1 - p)^{k-r}$
- Expected value: $\dfrac{r}{p}$
- Variance: $\dfrac{r\,(1 - p)}{p^2}$

- Probability generating function: $\left(\frac{p\,t}{1-(1-p)\,t}\right)^r$
- Moment generating function: $\left(\frac{p\,e^t}{1-(1-p)\,e^t}\right)^r$

Poisson distribution

This important distribution is often used to describe counts of number of events that occur within a fixed time or space unit. As such, it is the building block of the so-called Poisson process. For $\lambda > 15$, the Poisson probabilities are well approximated using the normal distribution. The binomial distribution with $n \to \infty$ and $np = \lambda$ fixed converges (in distribution) to a Poisson distribution with parameter λ.

- Parameter: $\lambda > 0$
- Values: $0, 1, \ldots$
- Probability mass function: $P(X = k) = e^{-\lambda}\,\frac{\lambda^k}{k!}$
- Expected value: λ
- Variance: λ
- Probability generating function: $e^{\lambda(t-1)}$
- Moment generating function: $e^{\lambda(e^t-1)}$

Uniform distribution (discrete)

The discrete uniform distribution should not be confused with the continuous uniform distribution. The uniform distributions are sometimes also called homogeneous distributions.

- Values: $0, 1, \ldots, n$
- Probability mass function: $P(X = k) = \frac{1}{n+1}$
- Expected value: $\frac{n}{2}$
- Variance: $\frac{n(n+2)}{12}$
- Probability generating function: $\frac{1-t^{n+1}}{(n+1)\,(1-t)}$
- Moment generating function: $\frac{1-e^{t(n+1)}}{(n+1)\,(1-e^t)}$

Appendix E

Continuous Probability Distributions

This appendix contains an overview of common continuous distributions, in alphabetical order. For more information on the distributions discussed in this chapter, we refer to [23] and [24]. Some expressions involve the Gamma function. This function is defined for positive x as

$$\Gamma(x) = \int_0^\infty e^{-t}\, t^{x-1}\, dt$$

Useful properties of the Gamma function are

- $\Gamma(n+1) = n!$ for non-negative integer n ($n \geq 0$)
- $\Gamma(x+1) = x\,\Gamma(x)$
- $\Gamma(\frac{1}{2}) = \sqrt{\pi}$

The second property also defines the Gamma function for negative, non-integer x.

Beta distribution

This distribution appears when studying the order statistics of a sample from a uniform random variable. If X is beta distributed with integer parameters α and β, then $P(X \leq t) = P(\alpha \leq Y \leq \alpha + \beta - 1)$, where Y is binomial with parameters $n = \alpha + \beta - 1$ and $p = t$.

- Parameters: $\alpha > 0$, $\beta > 0$
- Values: $(0, 1)$
- Density: $\dfrac{x^{\alpha-1}\,(1-x)^{\beta-1}}{B(\alpha,\beta)}$ where $B(\alpha,\beta)$ is the Beta function defined by

$$B(\alpha,\beta) := \frac{\Gamma(\alpha)\,\Gamma(\beta)}{\Gamma(\alpha+\beta)} = \int_0^1 y^{\alpha-1}\,(1-y)^{\beta-1}\, dy$$

- Expected value: $\dfrac{\alpha}{\alpha+\beta}$
- Variance: $\dfrac{\alpha\beta}{(\alpha+\beta+1)\,(\alpha+\beta)^2}$
- Characteristic function: $M(\alpha, \alpha+\beta, it)$, where M is a confluent hypergeometric function.

Cauchy distribution

The ratio of two independent normally distributed random variables with zero mean is Cauchy distributed. The Cauchy distribution with $\lambda = 1$ and $\theta = 0$ coincides with the Student t-distribution with one degree of freedom.

- Parameters: $\lambda > 0$, $-\infty < \theta < \infty$
- Values: $(-\infty, \infty)$
- Density: $\dfrac{1}{\pi\lambda\left[1 + \left(\dfrac{x-\theta}{\lambda}\right)^2\right]}$
- Expected value: does not exist
- Variance: does not exist
- Characteristic function: $e^{it\theta - |t|\lambda}$

χ^2-distribution

The χ^2-distribution is characterised by one parameter, denoted here by n, and known as the "degrees of freedom". Notation: χ^2_n. The name of the χ^2-distribution is derived from its relation to the standard normal distribution: if Z is a standard normal random variable, then its square $X = Z^2$ is χ^2 distributed, with one degree of freedom. If X_i are χ^2 distributed, and mutually independent, then the sum $X = \sum_i X_i$ is χ^2 and the parameter (degrees of freedom) is the sum of the parameters of the individual X_i. The χ^2-distribution is also a special case of the Gamma distribution, with $\alpha = \nu/2$ and $\lambda = 1/2$. The χ^2-distribution is of great importance in the Analysis of Variance (ANOVA), contingency table tests, and goodness-of-fit tests.

- Parameters: $\nu = 1, 2, \ldots$
- Values: $(0, \infty)$
- Density: $\dfrac{e^{-x/2}\, x^{(\nu-2)/2}}{2^{\nu/2}\, \Gamma(\nu/2)}$
- Expected value: ν
- Variance: 2ν
- Characteristic function: $(1 - 2it)^{-v/2}$

Erlang distribution

This is a special case of the Gamma distribution for positive integer values of α. It measures the time until the nth event in a Poisson process. If X_1 is Erlang distributed with parameters n and λ and if X_2 is Erlang distributed with parameters m and λ, and if X_1 and X_2 are independent, then $X_1 + X_2$ is Erlang distributed with parameters $n+m$ and λ. For $n = 1$, the Erlang distribution is the exponential distribution. If X_i are mutually independent and exponentially distributed with intensity λ, then $\sum_{i=1}^{n} X_i$ is Erlang distributed with parameters n and λ. Sometimes $\beta = 1/\lambda$ is used as parameter.

- Parameters: $n = 1, 2, \ldots, \lambda > 0$
- Values: $(0, \infty)$
- Density: $\dfrac{x^{n-1}\,\lambda^n\,e^{-\lambda\,x}}{(n-1)!}$
- Expected value: $\dfrac{n}{\lambda}$
- Variance: $\dfrac{n}{\lambda^2}$
- Characteristic function: $\left(1 - i\,\dfrac{t}{\lambda}\right)^{-n}$

Exponential distribution

This is a special case of both the Gamma and the Weibull distributions. The exponential distribution has the lack-of-memory property, in the sense that $P(X > s + t \mid X > s) = P(X > t)$. This property defines the exponential distribution, *i.e.*, no other continuous random variable has this property. The times between events in a Poisson process are exponentially distributed. If X_i are mutually independent and exponentially distributed with intensity λ, then $\sum_{i=1}^{n} X_i$ is Erlang distributed with parameters n and λ.

- Parameters: $\lambda > 0$; sometimes $\beta = 1/\lambda$ is used as parameter
- Values: $(0, \infty)$
- Density: $\lambda e^{-\lambda x}$
- Cumulative distribution function: $1 - e^{-\lambda x}$
- Expected value: $1/\lambda$
- Variance: $1/\lambda^2$
- Characteristic function: $\dfrac{1}{1 - it/\lambda}$

F-distribution

The F-distribution, named after the famous statistician Fisher, is the distribution of a ratio of two independent χ^2 random variables. It has two parameters, denoted by m and n, which are called the degrees of freedom of the numerator and the denominator, respectively. Notation: F_n^m. If X is Student t-distributed with n degrees of freedom, then X^2 is an F_n^1 variable. If U is χ^2 distributed with m degrees of freedom, V is χ^2 distributed with n degrees of freedom, and if U and V are independent, then $X = \frac{U/m}{V/n}$ is an F_n^m variable. The values $f_{n;\alpha}^m$ are defined by $P\left(F_n^m > f_{n;\alpha}^m\right) = \alpha$ (so they do not follow the customary definition of quantiles). From the definition of F_n^m as a ratio of two χ^2 variables, it follows that $f_{n;1-\alpha}^m = 1/f_{m;\alpha}^n$.

- Parameters: $m = 1, 2, \ldots, n = 1, 2, \ldots$
- Values: $(0, \infty)$

- Density: $\dfrac{\Gamma\left(\dfrac{m+n}{2}\right)}{\Gamma\left(\dfrac{m}{2}\right)\,\Gamma\left(\dfrac{n}{2}\right)}\,\dfrac{m^{m/2}\,n^{n/2}\,x^{(m/2)-1}}{(n+mx)^{(m+n)/2}}$
- Expected value: $\dfrac{n}{n-2}$ if $n \geq 3$; not defined for $n = 1$ or $n = 2$.
- Variance: $\dfrac{2n^2\,(m+n-2)}{m\,(n-2)^2\,(n-4)}$ $(n = 5, 6, \ldots)$
- Characteristic function: $M\left(\frac{1}{2}m; -\frac{1}{2}n; -\frac{n}{m}\,it\right)$, where M is a confluent hypergeometric function.

Gamma distribution

Special cases of the Gamma distribution include the χ^2-distribution ($\alpha = \nu/2$ and $\lambda = 1/2$), the Erlang distribution (α positive integer) and the exponential distribution ($\alpha = 1$).

- Parameters: $\alpha > 0$, $\lambda > 0$. Sometimes $\beta = 1/\lambda$ is used as parameter.
- Values: $(0, \infty)$
- Density: $\lambda^\alpha\,\dfrac{x^{\alpha-1}\,e^{-\lambda\,x}}{\Gamma(\alpha)}$
- Expected value: $\dfrac{\alpha}{\lambda}$
- Variance: $\dfrac{\alpha}{\lambda^2}$
- Characteristic function: $\left(1 - i\,\dfrac{t}{\lambda}\right)^{-\alpha}$

Gumbel distribution

The Gumbel distribution is one of the limiting distributions in extreme value theory.

- Parameters: $-\infty < \alpha < \infty$, $\beta > 0$
- Values: $(-\infty, \infty)$
- Cumulative distribution function: $e^{-e^{-(x-\alpha)/\beta}}$
- Expected value: $\alpha + \beta\gamma$ where $\gamma \approx 0,577216$ (Euler's constant)
- Variance: $\dfrac{\pi^2\,\beta^2}{6}$
- Characteristic function: $e^{i\alpha t}\,\Gamma(1 - i\beta t)$

Logistic distribution

This distribution is often used in the description of growth curves.

- Parameters: $-\infty < \alpha < \infty$, $\beta > 0$
- Values: $(-\infty, \infty)$
- Cumulative distribution function: $\left(1 + e^{-(x-\alpha)/\beta}\right)^{-1}$
- Expected value: α
- Variance: $\dfrac{\pi^2\,\beta^2}{3}$
- Characteristic function: $e^{i\alpha t}\,\dfrac{\pi\beta\,t}{\sinh \pi\beta t}$

Lognormal distribution

X has a lognormal distribution if $\ln X \sim N(\mu, \sigma^2)$.

- Parameters: $-\infty < \mu < \infty$, $\sigma > 0$
- Values: $(0, \infty)$
- Density: $\dfrac{1}{\sigma x\sqrt{2\pi}}\,e^{-\dfrac{(\ln x - \mu)^2}{2\sigma^2}}$
- Expected value: $e^{\mu + \frac{1}{2}\sigma^2}$
- Variance: $e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2}$
- Characteristic function: No closed expression known

Normal distribution

As suggested by its name, the normal distribution is the most important probability distribution in view of the Central Limit Theorem. Notation: $X \sim N(\mu, \sigma^2)$. The special case $\mu = 0$ and $\sigma = 1$ is called *standard normal distribution*, and a standard normal variable is most often denoted with the letter Z. The standard normal density is mostly written as $\varphi(z)$ and the cumulative distribution function as $\Phi(z)$. It holds that $\Phi(z) = 1 - \Phi(-z)$. The notation z_α is often defined as $P(Z > z_\alpha) = \alpha$ (so they do not follow the customary definition of quantiles).

- Parameters: $-\infty < \mu < \infty$, $\sigma > 0$
- Values: $(-\infty, \infty)$
- Density: $\dfrac{1}{\sigma\,\sqrt{2\pi}}\,e^{-\dfrac{(x-\mu)^2}{2\sigma^2}}$

- Expected value: μ
- Variance: σ^2
- Characteristic function: $e^{i\mu t} - (t^2\sigma^2/2)$

Pareto distribution

The Pareto distribution is often used in economical applications, such as the study of household incomes.

- Parameters: $a > 0$, $\theta > 0$
- Values: (a, ∞)
- Cumulative distribution function: $1 - \left(\frac{a}{x}\right)^{\theta}$
- Expected value: $\frac{\theta a}{\theta - 1}$ (if $\theta > 1$)
- Variance: $\frac{\theta a^2}{(\theta - 1)^2 (\theta - 2)}$ (if $\theta > 2$)
- Characteristic function: No closed expression known

Student t-distribution

If Z is a standard normal variable and U is a χ^2 variable with n degrees of freedom, and if Z and U are independent, then $\frac{Z}{\sqrt{U/n}}$ has a Student t-distribution with parameter n. Notation: T_n. The parameter is called the number of degrees of freedom. The standardised sample mean $\frac{\overline{X} - \mu}{S/\sqrt{n}}$ of a sample of normal random variables is Student t distributed with parameter $n - 1$.
The values $t_{n;\alpha}$ are defined by $P(T_n > t_{n;\alpha}) = \alpha$ (so they do not follow the customary definition of quantiles).
The Student t-distribution is named after the statistician William Gosset. His employer, the Guinness breweries, prohibited any scientific publication by its employees. Hence, Gosset published using a pen name, Student.

- Parameters: $n = 1, 2, \ldots$
- Values: $(-\infty, \infty)$
- Density: $\frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\,\Gamma\left(\frac{n}{2}\right)\left(1 + \frac{x^2}{n}\right)^{(n+1)/2}}$
- Expected value: 0 if $n \geq 2$, not defined for $n = 1$.
- Variance: $\frac{n}{n-2}$ ($n \geq 3$)

- Characteristic function: $\frac{1}{B(1/2, n/2)} \int_{-\infty}^{\infty} \frac{e^{itz\sqrt{n}}}{(1+z^2)^{(n+1)/2}}\, dz$, where $B(a,b)$ is the Beta function defined by $B(a,b) = \frac{\Gamma(a)\,\Gamma(b)}{\Gamma(a+b)} = \int_0^1 y^{a-1}\,(1-y)^{b-1}\, dy$.

Uniform distribution (continuous)

Also known as homogenous distribution. This distribution should not be confused with the discrete uniform distribution.

- Parameters: $-\infty < a < b < \infty$
- Values: (a, b)
- Density: $\frac{1}{b-a}$
- Cumulative distribution function: $\frac{x-a}{b-a}$
- Expected value: $\frac{a+b}{2}$
- Variance: $\frac{(b-a)^2}{12}$
- Characteristic function: $\frac{e^{itb} - e^{ita}}{i\,t\,(b-a)}$

Weibull distribution

The Weibull distribution often models survival times when the lack of memory property does not hold. The exponential distribution is a special case ($\beta = 1$ and $\lambda = 1/\delta$).

- Parameters: $\beta > 0$, $\delta > 0$
- Values: $(0, \infty)$
- Density: $\frac{\beta}{\delta}\left(\frac{x}{\delta}\right)^{\beta-1} e^{-(x/\delta)^{\beta}}$
- Expected value: $\delta\,\Gamma\left(1 + \frac{1}{\beta}\right)$
- Variance: $\delta^2\left[\Gamma\left(1+\frac{2}{\beta}\right) - \Gamma^2\left(1+\frac{1}{\beta}\right)\right]$
- Characteristic function: no closed expression known.

Index

References

[1] S. Asmussen, P.W. Glynn, *Stochastic simulation : algorithms and analysis*, Springer-Verlag, New York, USA, 2007.

[2] F. Augustin, A. Gilg, M. Paffrath, P. Rentrop, U. Wever: *Polynomial chaos for the approximation of uncertainties: Chances and limits*, Euro. Jnl. of Applied Mathematics, Vol. 19, pp. 149–190, 2008.

[3] L.J. Bain, M. Engelhardt: *Introduction to probability and mathematical statistics*, Duxbury, Belmont, California, 2nd edition, 1992.

[4] Th.G.J. Beelen, M. Kluitmans, M. Kole, E.J.W. ter Maten, J.M.F. Peters: *Living document statistics related analyses*, Version 0.7, NXP Semiconductors, PDM, 2007.

[5] L. Breiman, *Probability*, Series Classics in Applied Mathematics, Vol. 7 (Corrected reprint of the 1968 original), SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[6] J.A. Bucklew: *Introduction to Rare Event Simulation*, Springer-Verlag, New York, 2004.

[7] R.E. Caflisch: *Monte Carlo and quasi-Monte Carlo methods*, Acta Numerica, pp. 1–49, 1998.

[8] W. Cai, M.H. Kalos, M. de Koning, V.V. Bulatov: *Importance Sampling of rare events in Markov processes*, Physical Review E 66, 046703, pp. 1–13 (2002).

[9] B.H. Calhoun, A.P. Chandrakasan: *Static Noise Margin Variation for Sub-threshold SRAM in 65-nm CMOS*, IEEE J. of solid-state circuits 41-7, pp. 1673–1678.

[10] M.K. Cowles, B.P. Carlin: *Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review*, JASA Vol. 91, No. 434, pp. 883–904, 1996.

[11] F. den Hollander: *Large Deviations*, Fields Institute Monographs, AMS, Providence R.I., 2000.

[12] M. Denny: *Introduction to importance sampling in rare-events simulations*, Eur. J. Phys. 22, pp. 403–411, 2001.

[13] L. Devroye: *Nonuniform random variate generation*, Springer-Verlag, New York, USA, 1986.

[14] T.S. Doorn, R. Salters, L. Elvira Villagra: *SRAM design challenges – A research view on current status and future work*, Report NXP-TN-2007-00066, NXP Semiconductors, 2007.

[15] W. Feller: *An introduction to probability theory and its applications*, Volume 1, 3rd Edition, J. Wiley & Sons, 1970.

[16] E. Fix, J.L. Hodges, Jr.: *Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*, International Statistical Review / Revue Internationale de Statistique, Vol. 57, Nr. 3, pp. 238–247, 1989 (Reprint).
Original version: Report Nr. 4, Project Nr. 21-49-004, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[17] J.M. Hammersley, D.C. Hanscomb: *Monte Carlo methods*, Methuen, London, 1964.

[18] *NIST/SEMATECH e-Handbook of Statistical Methods*, http://www.itl.nist.gov/div898/handbook/.

[19] R. Haüßler, H. Kinzelbach: *Sensitivity-based stochastic analysis method for power variations*, Proc. Analog06, 2006.

[20] T.C. Hesterberg: *Advances in importance sampling*, PhD-Thesis Stanford Univ., 1988; added remarks.

[21] A.M. Johansen, L. Evers: *Monte Carlo methods*, Lecture Notes, Univ. of Bristol, UK, 2007.

[22] N.L. Johnson, S. Kotz, A.W. Kemp: *Univariate Discrete Distributions*, Wiley, New York, 1993.

[23] N.L. Johnson, S. Kotz, N. Balakrishnan: *Continuous Univariate Distributions volume 1*, 2^{nd} ed., Wiley, New York, 1994.

[24] N.L. Johnson, S. Kotz, N. Balakrishnan: *Continuous Univariate Distributions volume 2*, 2^{nd} ed., Wiley, New York, 1995.

[25] J. Kaipio, E. Somersaldo: *Statistical and computtaional inverse problems*, Springer Science+Busness Media, Inc., 2005.

[26] R. Kanj, R. Joshi, S. Nassif: *Mixture Importance Sampling and its application to the analysis of SRAM designs in the presence of rare failure events*, Proc. DAC 2006, 5.3, pp. 69–72, 2006.

[27] Y.B. Kim, D.S. Roh, M.Y. Lee: *Nonparametric adaptive importance sampling for rare event simulation*, In: J.A. Jones, R.R. Barton, K. Kang, P.A. Fishwick (Eds.): *Proceedings of the 2000 Winter Simulation Conference*, pp. 767–772, 2000.

[28] J.P.C. Kleijnen: *Design and analysis of simulation experiments*, Intern. Series in Operations Research and Management Science, Springer Science+Business Media, LLC, 2008.

[29] M. de Koning, W. Cai, B. Sadigh, T. Oppelstrup, M.H. Kalos, V.V. Bulatov: *Adaptive Importance sampling Monte Carlo simulation of rare transition events* The Journal of Chem. Physics 122, 074103, pp. 1–12, 2005.

[30] N.V. Korovkin, V.L. Chechurin, M. Hayakawa: *Inverse problems in electric circuits and electromagnetics*, Series Mathematical and Analytical techniques with Applications to Engineering, Springer Science+Business Media LLC, 2007.

[31] A. de Kraker: *Het toepassen van Importance Sampling bij Monte Carlo simulatie van rekenmodellen op statistische grondslag* (in Dutch), THE Report WE 82.11, Eindhoven Univ. of Technology, 1982.

[32] X. Li, J. Le, P. Gopalakrishnan, L.T. Pileggi: *Asymptotic Probability Extraction for Non-Normal Distributions of Circuit Performance*, IEEE Proc. ICCAD 2004, pp. 2–9, 2004.

[33] X. Li: *Statistical modeling, analysis and optimization for analog and RF ICs*, PhD-Thesis Carnegie Mellon University,2005.

[34] X. Li, J. Le, P. Gopalakrishnan, L.T. Pileggi: *Asymptotic Probability Extraction for Nonnormal Performance Distributions*, IEEE Trans. on Comp.-Aided Design of Integrated Circuits and Systems, Vol. 26, No. 1, pp. 16–37, 2007.

[35] W.L. Martinez, A.R. Martinez: *Computational statistics handbook with Matlab*, Chapman & Hall/CRC Press LLC, Boca Raton, FL, USA, 2002.

[36] M. Mascagni, A. Srinivasan: *Algorithm 806: SPRNG: A Scalable Library for Pseudorandom Number Generation*, ACM Trans. on Math. Software, Nr. 26, pp. 436–461, 2000.

[37] N. Mi, S.X.-D. Tan, P. Liu, J. Cui, Y. Cai, X. Hong: *Stochastic extended Krylov subspace method for variational analysis of on-chip power grid networks*, IEEE Proc. ICCAD 2007, pp. 48–53, 2007.

[38] H. Onoda, K. Miyashita, T. Nakayama, T. Kinoshita, H. Nishimura, A. Azuma, S. Yamada, F. Matsuoka: *0.7 V SRAM Technology with stress-enhanced dopant segregated Schottky (DSS) source/drain transistors for 32 nm node*, 2007 Symposium on VLSI Technology Digest of Technical Papers, 5B1, pp. 76-77, 2007.

[39] M. Paffrath, U. Wever: *Adapted polynomial chaos expansion for failure detection*, J. of Computational Physics, Vol. 226, pp. 263–281, 2007.

[40] M. Pagano, W. Sandmann: *Efficient rare event simulation: a tutorial on importance sampling*, HET-NETs '05, Int. Working Conf. on Performance Modelling and Evaluation of Heterogeneous Networks, Ilkley, West Yorkshire, U.K., 2005.

[41] J.M.F. Peters: *Statistical Analysis in Pstar*, Version 2.2, Philips Research, ED&T/Analogue Simulation, 2000.

[42] Pstar, Analogue circuit simulator of NXP Semiconductors.

[43] *Tolerance analysis with Pstar*, 4th Edition, Philips Research, ED&T/Analogue Simulation, 1996.

[44] A. Quarteroni, R. Sacco, F. Saleri: *Numerical mathematics, 2nd Ed.*, Springer Verlag, Berlin, 2006.

[45] D. Remondo, R. Srinivasan, V.F. Nicola, W.C. van Etten, H.E.P. Tattje: *Adaptive importance sampling for performance evaluation and parameter optimization of communication systems*, IEEE Trans. on Commun., Vol. 48-4, pp. 557-565, 2000.

[46] B.D. Ripley: *Stochastic simulation*, Wiley, New York, USA, 2006.

[47] E. Seevinck, F.J. List, J. Lohstroh: *Static noise margin analysis of MOS SRAM cells*, IEEE J. of Solid State Circuits, Vol. 22-5, pp. 748–754, 1987.

[48] P.J. Smith, M. Shafi, H. Gao: *Quick simulation: A review of Importance Sampling techniques in communication systems*, IEEE J. on Selected Areas in Comm., Vol. 15-4, pp. 597–613, 1997.

[49] Spectre, Analogue circuit simulator of Cadence Design Systems, Inc.

[50] *The Scalable Parallel Random Number Generators Library (SPRNG)*, http://sprng.cs.fsu.edu/.

[51] R. Srinivasan: *Importance Sampling - Applications in communications and detection*, Springer Verlag, (ISBN 3-540-43420-8), Berlin, 2002.

[52] J.F. Swidzinski, M. Keramat, K. Chang: *CAD Techniques for robust RF and wireless IC design*, Proc. 42nd Midwest Symposium on Circuits and Systems, Vol. 1, pp. 68-71, 1999.

[53] J.F. Traub, A.G. Werschulz: *Complexity and Information*, Cambridge University Press, 1998.

[54] D. Xiu: *Fast numerical methods for stochastic computations: A review*, Communicationsin Computational Physics, Vol. 5, Nr. 2-4, pp. 242–272, 2009.

PREVIOUS PUBLICATIONS IN THIS SERIES:

Number	Author(s)	Title	Month
09-33	J. Hulshof R. Nolet G. Prokert	Existence and linear stability of solutions of the ballistic VSC model	Oct. '09
09-34	F.A. Radu I.S. Pop	Simulation of reactive contaminant transport with non-equilibrium sorption by mixed finite elements and Newton method	Oct. '09
09-35	I.S. Pop B. Schweizer	Regularization schemes for degenerate Richards equations and outflow conditions	Oct. '09
09-36	A. Mikelić C.J. van Duijn	Rigorous derivation of a hyperbolic model for Taylor Dispersion	Nov. '09
09-37	E.J.W. ter Maten T.S. Doorn J.A. Croon A. Bargagli A. Di Bucchianico O. Wittich	Importance sampling for high speed statistical Monte-Carlo simulations	Nov. '09

Ontwerp: de Tantes,
Tobias Baanders, CWI