Efficient Estimation of the Expected Shortfall in a Nested Simulation Framework

M.J.K. van Wijngaarden

Delft University of Technology

Efficient Estimation of the Expected Shortfall

in a Nested Simulation Framework

by

M.J.K. van Wijngaarden

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday 25 February at 10:30 AM.

Student number:	4357418	
Project duration:	March 1, 2021 – February 25, 2022	
Thesis committee:	Dr. Ir. A. Papapantoleon,	TU Delft, supervisor
	Dr. Ir. J. Bierkens,	TU Delft

Abstract

We analyze three different methods that can approximate the expected shortfall of a financial portfolio in a nested simulation. In this simulation process, the outer simulation generates risk scenarios, and the inner simulation approximates the value of the financial portfolio under each risk scenario. The first method is the most standard one, and therefore we call it 'the standard Monte Carlo method'. This method uses the same amount of computational cost for each inner simulation. The second method adapts the computational cost of the inner simulation to the output of the outer simulation. Therefore, we call it 'the adaptive sampling method'. This technique has already been proven to work more efficiently than the standard Monte Carlo method [7]. The third method is called 'the multilevel Monte Carlo method' (MLMC) and is based on the technique that has been introduced by Giles [12]. This method approximates the expected shortfall multiple times and with different levels of accuracy. These estimators are used to give an accurate overall approximation of the expected shortfall. To the best of our knowledge, it has never been examined how efficiently the expected shortfall can be approximated in this manner by the MLMC method. This thesis will thoroughly explain how each method can be applied in a nested simulation to approximate the expected shortfall. In addition, an analysis is given on how to ensure that each method is used as efficiently as possible in the nested simulation. We also perform a numerical experiment in which we simulate a simplified version of a financial portfolio using stochastic processes. With the help of the numerical experiment, we examine which method is most efficient to approximate the expected shortfall.

Keywords: Expected shortfall, Financial portfolio, Nested simulation, Outer simulation, Inner simulation, Adaptive sampling, Multilevel Monte Carlo method

Preface

With this thesis, I complete my master in Applied Mathematics in the direction of Financial Engineering at the Technical University of Delft. This also marks the end of my time as a student, since working life is waiting.

I would like to thank the people that helped me through my thesis. First of all, my supervisor Antonis Papapantoleon, for the meetings and the information that contributed to this final result. Secondly, my friends from JC Ozon, roommates and others for the much-needed distraction that kept me somewhat sane. Last but not least, my family for their support during my thesis but also throughout my complete study duration.

M.J.K. van Wijngaarden
Rotterdam, February 2022

Contents

1
Introduction

For financial institutions the management and measurement of risks is of critical importance. Risk measurement aims to ensure that a financial institution has enough reserves relative to its investments. There are different types of risks in the financial sector. For example, a commercial bank can hold a portfolio of loans granted to various debtors. In this case, the bank is exposed to credit risk that some debtors will default. Another example is that an investor has a portfolio of shares for a certain period, where the portfolio's value evolves randomly during that time. The investor is then faced with market risks where the portfolio's value can drop below its original value. Due to the importance of those financial risks, financial institutions need to understand and analyze the risks of their operations. This would allow them to manage or control their risks. In most cases, when someone wants to understand their risks, one usually considers the loss of action. The loss can be modelled as a random variable with the help of probability measures. The only problem is that it can take on multiple values. This makes it hard to interpret. That is why risk measures were introduced, such that risks of an investment are easier to interpret. A risk measure is nothing more than a function that maps a random variable to a real number. The advantage of such a risk measure is that it is no longer needed to take a whole loss distribution into account. Instead, one only needs to focus on a single number [21]. The expected shortfall is the risk measure this thesis focuses on, which can be described as the expected return of an investment in the worst of cases.

In recent years, the bankruptcies of large and small banks have shown us that there is still room for improvement in the modelling and computation of risk measures. To approximate risk measures, one can model different scenarios, such that these scenarios can be used to compute the risk measures. The modelling of these scenarios is in most cases divided into two phases. In the first phase risk factors are sampled over a given time horizon, e.g. one day or one year from now. This first phase is called the outer stage. The second phase models the potential outcomes of a financial investment, which is conditional on each risk scenario. This phase is called the inner stage. This way of modelling is called a (two-level) nested simulation since it involves multiple levels of simulations. Such a nested simulation can represent a computational challenge since it can contain many steps to approximate risk measures [7]. This paper focuses on the algorithmic improvement of a nested simulation when approximating the expected shortfall.

The application of such a nested simulation scheme can be a big burden for financial institutions due to its high computational cost. An influential paper by Gordy and Juneja [30] analyzed the problem of estimating risk measures in a nested simulation. They considered multiple risk measures; the probability of a loss, the Value-at-Risk, and the expected shortfall. The goal was to divide the computational cost between the outer and the inner stages when minimizing the mean squared error of the resulting estimator. Their result showed that a relatively small number of inner stage samples should be used when approximating risk measures in a nested simulation, compared to the number of outer stage samples. Namely, the optimal ratio is to use $\mathcal{O}(\varepsilon^{-2})$ samples in the outer stage and $\mathcal{O}(\varepsilon^{-1})$ samples in the inner stage to obtain a mean squared error of $\mathcal{O}(\varepsilon^{2})$. The total computational cost is then equal to $\mathcal{O}(\varepsilon^{-3})$.

The approximation of risk measures can be improved in a nested simulation when we exploit the information that is considered by the risk measure itself. The outer stage samples that lead to the lowest returns play the most critical role in the approximation. With that information, one can adapt the number of inner stage samples conditional on the outcome of the outer stage samples. This could improve the complexity of the nested simulation. Adapting the number of inner stage samples to the outcome of the outer stage sample means that there is no longer a constant number of inner stage samples for each outer stage. That is why this is called non-uniform or adaptive sampling. This technique dates back to at least 2003, where Lee and Glynn [27] focussed on estimating the probability of a loss in a discrete case. Broadie, Du, and Moalemi [7] also tried to improve the complexity of estimating the probability of a loss, in a nested simulation setting. Their work showed that under certain circumstances adaptive sampling can bring the number of inner stage samples back to $\mathcal{O}(\varepsilon^{-1/2})$ samples instead of $\mathcal{O}(\varepsilon^{-1})$, to obtain a mean squared error of $\mathcal{O}(\varepsilon^{2})$ for the estimator. This would bring the total computational cost back to $\mathcal{O}(\varepsilon^{-5/2})$. Bouchard, Reghai, and Virrion [6] focused on the improvement of efficiency in the approximation of the expected shortfall, by also applying an adaptive number of inner stage samples in a nested simulation. Their result was that with adaptive sampling, the expected shortfall could be approximated with a smaller error using the same amount of computational cost compared to uniform sampling.

In parallel, the multilevel Monte Carlo (MLMC) method was introduced by Giles [12] in 2008, which is a method for computing expectations in complex simulations. An expanded overview can be found in [13]. The basic idea of the MLMC method is that relatively few computations are done with high accuracy at a high cost, and most computations are performed with low accuracy at low cost. It has been applied to a variety of simulations since its first introduction. These simulations range from financial stochastic differential equations (SDE's) [20] to stochastic partial differential equations (SPDE's) for biological and non-linear systems [28, 31]. The MLMC method can be applied to a simulation when there are parameters that control the computational cost and accuracy of the estimator. Since a nested simulation contains inner and outer stage samples that regulate the accuracy and computational cost of the estimator, the MLMC method can be applied in a nested simulation. Therefore, it is no surprise that the MLMC method is already used to improve the approximation of risk measures in a nested simulation. Giles and Haji-Ali [14, 15] applied the MLMC method for the approximation of the Value-at-Risk and the expected shortfall in a nested simulation. Their research also contained the adaptive sampling technique, based on the work of Broadie, Du, and Moalemi [7]. Giles and Haji-Ali combined the MLMC method and adaptive sampling to approximate the Value-at-Risk of a portfolio. Next, was the expected shortfall approximated with the use of the MLMC method. The only problem of applying a combination of adaptive sampling and the MLMC method for the approximation of the Value-at-Risk, was that Broadie, Du, and Moalemi only focused on the probability of a loss and not the Value-at-Risk. Giles and Haji-Ali solved this problem by adding an extra stochastic root-finding technique in their algorithm. Consequently, adding that root-finding algorithm technique resulted in an increase in the computational cost. Another disadvantage of the approach by Giles and Haji-Ali, is that for the approximation of the expected shortfall, the MLMC method must be applied twice. That is because the Value-at-Risk must first be approximated with the MLMC method before the expected shortfall can be approximated. Our approach is therefore slightly different. We apply the MLMC method in our simulation to make multiple approximations of the expected shortfall, with varying levels of accuracy and amount of computational cost. These estimators of the expected shortfall are used to make an overall approximation of the expected shortfall. It has already been demonstrated that this approach is more efficient than the standard Monte Carlo method for the approximation of the Value-at-Risk [1].

This thesis sets up three methods that can approximate the expected shortfall. The first one uses a constant number of inner stage samples for each outer stage sample. The second one applies adaptive sampling in the nested simulation, and the third uses the MLMC method. Adaptive sampling and the MLMC method are relatively new methods that aim to estimate random variables efficiently. To date, we are not aware of any published articles that used the MLMC method in a nested simulation to obtain multiple estimates with different accuracies of the expected shortfall, by which the expected shortfall is approximated. In addition, it has never been examined to our knowledge whether the MLMC method or adaptive sampling is more efficient for approximating the expected shortfall in a nested simulation. So we aim to implement each method as efficiently as possible in the nested simulation and examine which method is most efficient.

The following sections are organized as follows. Section 2 focuses on the theory behind risk measures, their origin, and the properties of the expected shortfall. The simulation framework is explained in section 3, which clarifies how a nested simulation works and how the expected shortfall can be approximated with the outcomes of a simulation. The application of the standard Monte Carlo in a nested simulation is illustrated in section 4. Section 5 explains how we apply adaptive sampling in a nested simulation. In section 6 is described how the MLMC method works and how we apply it in our simulation. Section 7 shows with the help of numerical experiments that the theories that are stated for each method also apply in practice. In section 8, we test the efficiency of each method with the help of numerical experiments. The conclusion of this thesis and the possible follow-up research is stated in section 9.

2
Risk measures

In this section, we discuss different examples of downside risk measures. This can provide insight into how the risk measures have developed in recent years. We also explain why we have decided to estimate the expected shortfall and show the useful properties for this risk measure.

When financial institutions make their decisions, they do not only take into account the potential profits they could make, but also their potential losses. Since these potential losses are of great urgency to an investor, is it important to identify and understand these risks, so that they can be controlled or managed. To understand risk, one often considers the loss of an activity. With the use of probability measures, the loss can be modeled as a random variable. The only problem is that a random variable can have multiple values and is therefore difficult to interpret. For this reason risk measures were invented. A risk measure is a function that maps a random variable to a real number. Using risk measures, one only needs to focus on a number instead of the whole loss distribution. Many risk measures have been introduced and used by various financial industries. We will focus on the downside risk measures, since these risk measures target the worst case scenarios of an investment.

2.1. Downside risk measures

In this section we present different examples of downside risk measures. A risk measure is defined as follows, let $\mathcal{R}(X)$ be a risk measure where X is a random outcome from a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We define X as a random return of some portfolio or asset. The risk measure is defined as a mapping $\mathcal{R} : \mathcal{X} \to \mathbb{R}$. Here is $\mathcal{X}$ a linear space of the $\mathcal{F}$-measurable functions $X : \Omega \to \mathbb{R}$. We may assume in a more general setting that $\mathcal{X}$ is a locally convex space, where we have that $\mathcal{X} = \mathcal{L}_p(\Omega, \mathcal{F}, \mathbb{P})$, for $1 \leq p \leq \infty$. The value of p is in most cases clear from the context. The expectation is denoted by $\mathbb{E}[\cdot]$ with respect to $\mathbb{P}$.

The start of modern portfolio analysis was laid out by Markowitz in 1952, who proposed to identify the risk of the portfolio with the volatility of its return. In doing so, the work of Markowitz led to the formalization of the basic view that any decision under uncertainty can be evaluated in terms of its risk and reward. Markowitz's pioneering ideas are still widely used today in many areas of decision-making. The shortcomings in using volatility as a risk measure have been noted by Markowitz himself. With volatility, both the positive and negative deviations from the expected value are included, while in risk measurement we are only interested in the negative deviations. He therefore proposed in 1959 the semi variance σ_-^2, for a more accurate estimate of risk exposure, which is defined as

$$\sigma_-^2 = \|(X - \mathbb{E}[X])_-\|_2^2, \tag{2.1}$$

with $x_- = \max(0, -x)$ and $\|X\|_2 = \left(\mathbb{E}[|X|^2]\right)^{1/2}$. The semi variance only takes into account the negative deviation from the expected value and does not include the positive deviation, it measures how large the deviation in losses is. With the semi variance, the risk of X is only associated with X falling below its expected level $\mathbb{E}[X]$, whereas in many applications it is desirable to estimate the average shortfall below a reference value $a \in \mathbb{R}$. This is exactly what the risk measure Expected Regret (ER) determines, which is defined as

$$\mathrm{ER}(X) = \mathbb{E}[(a - X)_+] = \mathbb{E}[(X - a)_-]. \tag{2.2}$$

The ER falls under a larger group of risk measures called lower partial moment (LPM), the start of these risk measures in portfolio theory has mainly been driven by Bawa in 1975 [4] and by Fishburn in 1977 [11]. The LPM risk measures are defined as

$$\mathrm{LPM}_p(X, a) = \mathbb{E}\Big[(X-a)_-^p\Big], \tag{2.3}$$

where $p \geq 0$. For $p = 0$ the LPM can be considered as the risk measure probability for loss, since we have that $\mathbb{E}\Big[(X-a)_-^0\Big] = \mathbb{E}\big[\mathbf{1}_{\{X \leq a\}}\big] = \mathbb{P}(X \leq a)$. Another widely used risk measure is the worst case scenario. The worst case scenario, also known as the maximum loss, determines the maximum loss that can occur over a given time and is defined as

$$\mathrm{WCR}(X) = -\sup\{x \in \mathbb{R} : \mathbb{P}(X \geq x) = 1\}. \tag{2.4}$$

The WCR is applied in many decision-making problems under uncertainty, including machine scheduling, network problems and portfolio optimization [23, 35]. The reason why WCR is often used in practice is that it is easy to interpret and also efficient to implement in stochastic programming models. During the last 20 years the Value-at-Risk (VaR) is one of the most popular ways for financial institutions to assess their risks. Delbaen [10] described the VaR at level α (VaR$_\alpha$), as the smallest value for which the probability that $-X$ does not exceed the VaR$_\alpha$ is at least 1-α. To define this mathematically, we first define the following quantiles

$$x_{(\alpha)} = q_\alpha(X) = \inf\{x \in \mathbb{R} :\ \mathbb{P}[X \leq x] \geq \alpha\} \quad \text{(the lower } \alpha \text{ quantile)} \tag{2.5}$$

$$x^{(\alpha)} = q^\alpha(X) = \inf\{x \in \mathbb{R} :\ \mathbb{P}[X \leq x] > \alpha\} \quad \text{(the upper } \alpha \text{ quantile)} \tag{2.6}$$

Next we can define the VaR$_\alpha$ as follows,

$$\mathrm{VaR}_\alpha(X) = -x^{(\alpha)} = q_{1-\alpha}(-X). \tag{2.7}$$

Due to its many applications by banking institutions and its intuitive definition, the VaR is seen as a standard measurement for the risk exposure of financial positions. However, the VaR has some shortcomings. For example, the VaR does not take into account the extreme losses above the α-quantile level. Another shortcoming is that the VaR of a financial portfolio that consist of multiple components, may be greater than the sum of all the VaR's of each component. This is because the VaR is a non-convex function of X. Thereby, the VaR$_\alpha$ is not continuous with respect to α. This means that small changes in α can lead to significant jumps in the VaR's risk estimates. A new risk measure was proposed to alleviate the problems inherent in VaR, this risk measure is called the expected shortfall and is also known in the literature as the Conditional Value-at-Risk (CVaR). The expected shortfall can be defined with confidence level $\alpha \in (0, 1)$, as the conditional expectation of loss that exceeds the VaR$_\alpha$ and is defined by Acerbi [2] as

$$\mathrm{ES}_\alpha(X) = -\frac{1}{\alpha}\Big(\mathbb{E}[X\mathbf{1}_{\{X \leq x_{(\alpha)}\}}] + x_{(\alpha)}(\alpha - \mathbb{P}[X \leq x_{(\alpha)}])\Big), \tag{2.8}$$

In addition to being convex, the expected shortfall is also continuous in α. As a result, small variations in α will result in small changes in the risk estimates of the expected shortfall, which is useful in risk management. Other definitions for the expected shortfall are often used in the literature than the one in (2.8), which will be discussed in section 2.3.

2.2. Coherent risk measures

The formulation of risk measures has been given an axiomatic formulation by the works of Artzner et al. in 1999 and by Delbaen in 2002. They developed an axiomatic understanding for the concept of risk measures, by creating the notion of coherent risk measures. There is a variety of approaches to define and estimate risks.

In the development of risk measures, it is often the case that these measures are designed for uncertainty in decision making under certain circumstances. These measures have therefore attractive properties for those situations, but also lack fundamental features making them inapplicable in many other situations. A good example of this is the VaR, which has often been criticized both in practice and by academics for its lack of convexity and its other shortcomings. Artzner et al. and Delbaen therefore proposed a construction of risk measures with an axiomatic approach. The idea is that the axioms are used to determine what a well-defined risk function must satisfy. There are four axioms, and we call the risk measures that satisfy these four axioms coherent risk measures.

A mapping $\mathcal{R}: \mathcal{X} \to \mathbb{R}$ is called a coherent risk measure if it satisfies the following conditions:

$$\begin{aligned}
&\text{(A1) Monotonicity}: \; X \geq 0 \implies \mathcal{R}(X) \leq 0 \quad \forall X \in \mathcal{X}\\
&\text{(A2) Subadditivity}: \; \mathcal{R}(X+Y) \leq \mathcal{R}(X) + \mathcal{R}(Y) \quad \forall X, Y \in \mathcal{X}\\
&\text{(A3) Positive homogeneity}: \; \mathcal{R}(\lambda X) = \lambda \mathcal{R}(X) \quad \forall X \in \mathcal{X}, \lambda > 0\\
&\text{(A4) Translation invariance}: \; \mathcal{R}(X+a) = \mathcal{R}(X) - a \quad \forall X \in \mathcal{X}, a \in \mathbb{R}
\end{aligned}$$

An important property of a coherent risk measure is Fatou's property. Fatou's property is that for every bounded sequence $\{X_n\}$, which almost surely converges to X, holds that $\mathcal{R}(X) \leq \lim_{n\to\infty} \inf \mathcal{R}(X_n)$. Another important property that can be formed from axioms (A1), (A2), and (A3), is that $\mathcal{R}(X) \leq \mathcal{R}(Y)$ when $X \geq Y$, also that $X \geq -a$ implies $\mathcal{R}(X) \leq a$ for all $a \in \mathbb{R}$. Axiom (A2) is useful for risk reduction through diversification. With the help of axiom (A3), one can describe axiom (A2) as, (A2') convexity: $\mathcal{R}(\lambda X + (1-\lambda)Y) \leq \lambda\mathcal{R}(X) + (1-\lambda)\mathcal{R}(Y)$ for all $X, Y \in \mathcal{X}$ and $\lambda \in [0,1]$. From a mathematical point of view axiom (A2') is particularly useful, because this allows the risk to be minimized efficiently over a convex set. Axiom (A4) is logical from a financial interpretation, if X is the payoff of a financial position, then the risk can be reduced by the same amount of cash that is added, so $\mathcal{R}(X + \mathcal{R}(X)) = 0$. If we then combine axiom (A4) with (A3), we get $\mathcal{R}(0) = 0$ and $\mathcal{R}(a) = -a$ for all $a \in \mathbb{R}$. This is again logical from the financial point of view.

With the use of a coherent risk measure $\mathcal{R}$, the acceptance set $\mathcal{A}_\mathcal{R}$ is defined as a convex cone, such that

$$\mathcal{A}_\mathcal{R} = \{X \in \mathcal{X} | \mathcal{R}(X) \leq 0\}. \tag{2.9}$$

Using the acceptance set $\mathcal{A}_\mathcal{R}$, $\mathcal{R}$ can be described as $\mathcal{R}(X) = \inf\{c \in \mathbb{R} | X + c \in \mathcal{A}_\mathcal{R}\}$.

Well-known examples of risk measures that fulfill the coherence properties are the maximum loss and the expected shortfall.

2.3. The expected Shortfall

As mentioned earlier, the VaR is one of the most popular ways to assess risks for financial institutions in the past 20 years. The financial crisis of 2008 showed us that there are still shortcomings in the financial risk management methodologies. The Basel committee of banking supervision of the Bank for International Settlements stated that "it is questionable whether it (VaR) meets the objectives of prudential regulation which seeks to ensure that banks have sufficient capital to survive low probability, or 'tail' events." and "a number of weaknesses have been identified with using VaR for determining regulatory capital requirements, including its inability to capture 'tail risk,'" [17]. They also suggested to replace the VaR with the expected shortfall when assessing financial risks. With all this taken into account, the focus of our thesis is on the expected shortfall.

The expected shortfall has properties that are beneficial for a risk measure. Examples of those beneficial properties are the continuity of the expected shortfall in α and that it is a coherent risk measure. In addition, the monotonicity of α is also an important property of the expected shortfall, which means that the risk will get greater when α gets smaller. We show that these properties apply to the expected shortfall. When proving these properties, we observe that there are other ways to define the expected shortfall, as is stated in (2.8). Most studies that also focus on the expected shortfall often use these other mathematical definitions.

The following notations are used when we prove the properties of the expected shortfall:

$$\overline{x}_{(\alpha)} = -\mathrm{ES}_\alpha(X) = \frac{1}{\alpha}\left(\mathbb{E}[X\mathbf{1}_{\{X \leq x_{(\alpha)}\}}] + x_{(\alpha)}(\alpha - \mathbb{P}[X \leq x_{(\alpha)}])\right) \tag{2.10}$$

$$X^+ = \begin{cases} X, \text{ if } X > 0 \\ 0, \text{ if } X \leq 0 \end{cases} \text{, and its negitive part by } X^- = (-X)^+ \tag{2.11}$$

We start with proposition 2.3.1, the monotonicity of the expected shortfall in α.

Proposition 2.3.1 If X is a random variable with $\mathbf{E}[X^-] < \infty$, then for any $\alpha \in (0,1)$ and any $\varepsilon > 0$ with $\alpha + \varepsilon < 1$, we have the following inequality

$$\mathrm{ES}_{\alpha+\varepsilon}(X) \leq \mathrm{ES}_\alpha(X) \tag{2.12}$$

Proof. First we define $\mathbf{1}^{(\alpha)}_{\{X\leq x\}}$ for $x \in \mathbb{R}$ as follows,

$$\mathbf{1}^{(\alpha)}_{\{X\leq x\}} = \begin{cases} \mathbf{1}_{\{X\leq x\}}, & \text{if } \mathbb{P}(X=x)=0 \\ \mathbf{1}_{\{X\leq x\}} + \frac{\alpha-\mathbb{P}(X\leq x)}{\mathbb{P}(X=x)}\mathbf{1}_{\{X=x\}}, & \text{if } \mathbb{P}(X=x)>0 \end{cases}. \tag{2.13}$$

Next, we show that the following equality's hold with the help of notations (2.5),(2.10) and (2.13),

$$\mathbb{E}\left[\mathbf{1}^{(\alpha)}_{\{X\leq x_{(\alpha)}\}}\right] = \alpha, \quad \text{and} \quad \frac{1}{\alpha}\mathbb{E}\left[X\mathbf{1}^{(\alpha)}_{\{X\leq x_{(\alpha)}\}}\right] = \overline{x}_{(\alpha)}. \tag{2.14}$$

The equality's in (2.14) are easy to verify when the distribution of X is continuous because then holds that $\mathbb{P}\big(X \leq x_{(\alpha)}\big) = \alpha$, since $x_{(\alpha)}$ is defined as $\inf\{x \in \mathbb{R} : \mathbb{P}[X \leq x] \geq \alpha\}$.

Suppose X has a discrete distribution, then it holds that $\mathbb{P}\left(X = x_{(\alpha)}\right) > 0$. So we have that,

$$\begin{aligned} \mathbb{E}\left[\mathbf{1}^{(\alpha)}_{\{X\leq x_{(\alpha)}\}}\right] &= \mathbb{E}\left[\mathbf{1}_{\{X\leq x_{(\alpha)}\}} + \frac{\alpha - \mathbb{P}\left(X \leq x_{(\alpha)}\right)}{\mathbb{P}\left(X = x_{(\alpha)}\right)}\mathbf{1}_{\{X=x_{(\alpha)}\}}\right] \\ &= \mathbb{P}\left(X \leq x_{(\alpha)}\right) + \frac{\alpha - \mathbb{P}\left(X \leq x_{(\alpha)}\right)}{\mathbb{P}\left(X = x_{(\alpha)}\right)}\mathbb{P}\left(X = x_{(\alpha)}\right) \\ &= \alpha \end{aligned} \tag{2.15}$$

and

$$\begin{aligned} \frac{1}{\alpha}\mathbb{E}\left[X\mathbf{1}^{(\alpha)}_{\{X\leq x_{(\alpha)}\}}\right] &= \frac{1}{\alpha}\mathbb{E}\left[X\left(\mathbf{1}_{\{X\leq x_{(\alpha)}\}} + \frac{\alpha - \mathbb{P}\left(X \leq x_{(\alpha)}\right)}{\mathbb{P}\left(X = x_{(\alpha)}\right)}\mathbf{1}_{\{X=x_{(\alpha)}\}}\right)\right] \\ &= \frac{1}{\alpha}\left(\mathbb{E}\left[X\mathbf{1}_{\{X\leq x_{(\alpha)}\}}\right] + \frac{\alpha - \mathbb{P}\left(X \leq x_{(\alpha)}\right)}{\mathbb{P}\left(X = x_{(\alpha)}\right)}\mathbb{E}\left[X\mathbf{1}_{\{X=x_{(\alpha)}\}}\right]\right) \\ &= \frac{1}{\alpha}\left(\mathbb{E}\left[X\mathbf{1}_{\{X\leq x_{(\alpha)}\}}\right] + \frac{\alpha - \mathbb{P}\left(X \leq x_{(\alpha)}\right)}{\mathbb{P}\left(X = x_{(\alpha)}\right)}x_{(\alpha)}\mathbb{P}\left(X = x_{(\alpha)}\right)\right) \\ &= \overline{x}_{(\alpha)}. \end{aligned} \tag{2.16}$$

The inequality in (2.12) is proved as follows:

$$\begin{aligned} \mathrm{ES}_\alpha(X) - \mathrm{ES}_{\alpha+\varepsilon}(X) &= \overline{x}_{(\alpha+\varepsilon)} - \overline{x}_{(\alpha)} \\ &= \mathbb{E}\left[X\left((\alpha+\varepsilon)^{-1}\mathbf{1}^{(\alpha+\varepsilon)}_{\{X\leq x_{(\alpha+\varepsilon)}\}} - \alpha^{-1}\mathbf{1}^{(\alpha)}_{\{X\leq x_{(\alpha)}\}}\right)\right] \\ &= \frac{1}{\alpha(\alpha+\varepsilon)}\mathbb{E}\left[X\left(\alpha\mathbf{1}^{(\alpha+\varepsilon)}_{\{X\leq x_{(\alpha+\varepsilon)}\}} - (\alpha+\varepsilon)\mathbf{1}^{(\alpha)}_{\{X\leq x_{(\alpha)}\}}\right)\right] \\ &\geq \frac{1}{\alpha(\alpha+\varepsilon)}\mathbb{E}\left[x_{(\alpha)}\left(\alpha\mathbf{1}^{(\alpha+\varepsilon)}_{\{X\leq x_{(\alpha+\varepsilon)}\}} - (\alpha+\varepsilon)\mathbf{1}^{(\alpha)}_{\{X\leq x_{(\alpha)}\}}\right)\right] \\ &= \frac{x_{(\alpha)}}{\alpha(\alpha+\varepsilon)}\left(\alpha\mathbb{E}\left[\mathbf{1}^{(\alpha+\varepsilon)}_{\{X\leq x_{(\alpha+\varepsilon)}\}}\right] - (\alpha+\varepsilon)\mathbb{E}\left[\mathbf{1}^{(\alpha)}_{\{X\leq x_{(\alpha)}\}}\right]\right) \\ &= \frac{x_{(\alpha)}}{\alpha(\alpha+\varepsilon)}(\alpha(\alpha+\varepsilon) - \alpha(\alpha+\varepsilon)) \\ &= 0 \qquad \square \end{aligned} \tag{2.17}$$

Proposition 2.3.2 states the continuity of the expected shortfall in α.

Proposition 2.3.2 If X is a random variable with $\mathbf{E}[X^-] < \infty$, then the mapping $\alpha \to \mathrm{ES}_\alpha$ is continuous on (0,1).

Proof. Let X be a random variable on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with $\mathbf{E}[X^-] < \infty$ and $\alpha \in (0,1)$. When proving proposition 2.3.2 we show that the following equation holds:

$$\overline{x}_{(\alpha)} = \frac{1}{\alpha}\int_0^\alpha x_{(y)}\,\mathrm{d}y \tag{2.18}$$

Let Y be a random variable on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, that is uniformly distributed on (0,1), i.e. $\mathbb{P}(Y \leq y) = y$, for $y \in (0,1)$. Then has the random variable $Z = x_{(Y)}$ the same distribution as X. We obtain the following properties since $y \rightarrow x_{(y)}$ is non-decreasing

$$\begin{aligned} \{Y \leq \alpha\} &\subset \{Z \leq x_{(\alpha)}\} && (2.19) \\ \{Y > \alpha\} \cap \{Z \leq x_{(\alpha)}\} &\subset \{Z = x_{(\alpha)}\} && (2.20) \end{aligned}$$

With the use of these properties in (2.19) and (2.20), we have that:

$$\begin{aligned} \int_0^\alpha x_{(y)}\,\mathrm{d}y &= \mathbb{E}[Z\mathbf{1}_{\{Y\leq\alpha\}}] \\ &= \mathbb{E}[Z\mathbf{1}_{\{Z\leq x_{(\alpha)}\}}] - \mathbb{E}[Z\mathbf{1}_{\{Y>\alpha\}\cap\{Z\leq x_{(\alpha)}\}}] \\ &= \mathbb{E}[Z\mathbf{1}_{\{X\leq x_{(\alpha)}\}}] + x_{(\alpha)}\left(\alpha - \mathbb{P}(X \leq x_{(\alpha)})\right) && (2.21) \end{aligned}$$

We have that the equation in (2.18) holds when we divide by α. Proposition 2.3.2 follows immediately from the fact that equation (2.18) holds and that $\mathrm{ES}_\alpha(X) = -\overline{x}_{(\alpha)}$. □

The continuity of the expected shortfall in α ensures that it does not change dramatically when there is a small change in the confidence level α. The proof of proposition 2.3.2 shows something else interesting, namely equation (2.18). This allows us to define the expected shortfall as follows

$$\mathrm{ES}_\alpha(X) = -\frac{1}{\alpha}\int_0^\alpha q_y(X)\,\mathrm{d}y \tag{2.22}$$

Most studies that focus on the expected shortfall use this as its definition. Intuitively, this definition is also easier to understand than the one that is stated in (2.8).

Finally, proposition 2.3.3 states that the expected shortfall is a coherent risk measure.

Proposition 2.3.3 Let $\mathcal{X}$ be a set of random variables on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that $\mathbb{E}[X^-] < \infty$ for all $X \in \mathcal{X}$. Then $\mathrm{ES}_\alpha(X): \mathcal{X} \rightarrow \mathbb{R}$ for $X \in \mathcal{X}$ and for a fixed $\alpha \in (0,1)$ satisfies the following conditions

$$\text{(i) Monotonicity}: \ X \in \mathcal{X},\ X \geq 0 \implies \mathrm{ES}_\alpha(X) \leq 0$$

$$\text{(ii) Subadditivity}: \ X, Y \in \mathcal{X},\ X+Y \in \mathcal{X} \implies \mathrm{ES}_\alpha(X+Y) \leq \mathrm{ES}_\alpha(X) + \mathrm{ES}_\alpha(Y)$$

$$\text{(iii) Positive homogeneity}: \ X \in \mathcal{X},\ \lambda > 0\ \lambda X \in \mathcal{X} \implies \mathrm{ES}_\alpha(\lambda X) = \lambda \mathrm{ES}_\alpha(X)$$

$$\text{(iv) Translation invariance}: \ X \in \mathcal{X},\ a \in \mathbb{R} \implies \mathrm{ES}_\alpha(X+a) = \mathrm{ES}_\alpha(X) - a$$

Proof. Condition (i) monotonicity: Let $X \in \mathcal{X}$ be a random variable with $\mathbb{E}[X^-]$ and $X \geq 0$, the following inequality holds for all $\alpha \in (0,1)$:

$$x_{(\alpha)} = \inf\{x \in \mathbb{R}: \ \mathbb{P}(X \leq x) \geq \alpha\} \geq 0 \tag{2.23}$$

Since this holds for all $\alpha \in (0,1)$, do we have that:

$$\overline{x}_{(\alpha)} = \frac{1}{\alpha}\int_0^\alpha x_{(y)}\,\mathrm{d}y \geq 0 \tag{2.24}$$

The expected shortfall is defined as $\mathrm{ES}_\alpha(X) = -\overline{x}_{(\alpha)}$, so we have that $\mathrm{ES}_\alpha(X) \leq 0$.

Condition (ii) subadditivity: Let $X, Y \in \mathcal{X}$ be two random variables with $\mathbb{E}[X^-] < \infty$ and $\mathbb{E}[Y^-] < \infty$. We define $Z = X + Y$, then by using equations (2.14) we have for all $\alpha \in (0,1)$ that:

$$\begin{aligned}\alpha\left(\mathrm{ES}_\alpha(X)+\mathrm{ES}_\alpha(Y)-\mathrm{ES}_\alpha(Z)\right) &= \mathbb{E}\left[-X\mathbf{1}^{(\alpha)}_{\{X\le x_{(\alpha)}\}} - Y\mathbf{1}^{(\alpha)}_{\{Y\le y_{(\alpha)}\}} + Z\mathbf{1}^{(\alpha)}_{\{Z\le z_{(\alpha)}\}}\right] \\ &= \mathbb{E}\left[X\left(\mathbf{1}^{(\alpha)}_{\{Z\le z_{(\alpha)}\}} - \mathbf{1}^{(\alpha)}_{\{X\le x_{(\alpha)}\}}\right) + Y\left(\mathbf{1}^{(\alpha)}_{\{Z\le z_{(\alpha)}\}} - \mathbf{1}^{(\alpha)}_{\{Y\le y_{(\alpha)}\}}\right)\right] \\ &\ge x_{(\alpha)}\mathbb{E}\left[\mathbf{1}^{(\alpha)}_{\{Z\le z_{(\alpha)}\}} - \mathbf{1}^{(\alpha)}_{\{X\le x_{(\alpha)}\}}\right] + y_{(\alpha)}\mathbb{E}\left[\mathbf{1}^{(\alpha)}_{\{Z\le z_{(\alpha)}\}} - \mathbf{1}^{(\alpha)}_{\{Y\le y_{(\alpha)}\}}\right] \\ &= x_{(\alpha)}(\alpha-\alpha) + y_{(\alpha)}(\alpha-\alpha) \\ &= 0 \qquad (2.25)\end{aligned}$$

Thus, it follows that $\mathbb{E}(X+Y) \ge \mathbb{E}(Z)$, which proves the subadditivity.

Condition (iii) positive homogeneity: Let $\lambda > 0$ and $X \in \mathcal{X}$ be a random variable with $\mathbb{E}[X^-]$, then holds the following equality for all $\lambda \in (0,1)$:

$$q(\lambda X) = \inf\{x\in\mathbb{R} : \mathbb{P}[\lambda\cdot X \le x] \ge \alpha\} = \lambda\cdot\inf\{x\in\mathbb{R} : \mathbb{P}[X\le x]\ge\alpha\} = \lambda q(X). \qquad (2.26)$$

So we have that:

$$\begin{aligned}\mathrm{ES}_\alpha(\lambda X) &= -\frac{1}{\alpha}\int_0^\alpha q_y(\lambda X)\,\mathrm{d}y \\ &= -\frac{1}{\alpha}\int_0^\alpha \lambda q_y(X)\,\mathrm{d}y \\ &= -\frac{\lambda}{\alpha}\int_0^\alpha q_y(X)\,\mathrm{d}y \\ &= \lambda\mathrm{ES}_\alpha(X) \qquad (2.27)\end{aligned}$$

Condition (iv) translation invariance: Let $a\in\mathbb{R}$ and $X\in\mathcal{X}$ be a random variable with $\mathbb{E}[X^-]$, then holds the following equality for all $a\in\mathbb{R}$:

$$q(X+a) = \inf\{x\in\mathbb{R} : \mathbb{P}[X+a\le x]\ge\alpha\} = \inf\{x\in\mathbb{R} : \mathbb{P}[X\le x]\ge\alpha\} + a = q(X)+a. \qquad (2.28)$$

This gives us the following:

$$\begin{aligned}\mathrm{ES}_\alpha(X+a) &= -\frac{1}{\alpha}\int_0^\alpha q_y(X+a)\,\mathrm{d}y \\ &= -\frac{1}{\alpha}\int_0^\alpha q_y(X)+a\,\mathrm{d}y \\ &= -\frac{1}{\alpha}\left(\int_0^\alpha q_y(X)\,\mathrm{d}y + \int_0^\alpha a\,\mathrm{d}y\right) \\ &= -\frac{1}{\alpha}\left(\int_0^\alpha q_y(X)\,\mathrm{d}y + a\alpha\right) \\ &= -\frac{1}{\alpha}\int_0^\alpha q_y(X)\,\mathrm{d}y - a \\ &= \mathrm{ES}_\alpha(X) - a. \qquad \square \qquad (2.29)\end{aligned}$$

3

Simulation framework

In this section, we discuss how a nested simulation exactly works. In addition, we discuss how the simulated returns can be used to estimate the expected shortfall. Finally, we explain how we simulate the return of the portfolio in this thesis and how the value of the expected shortfall can be computed analytically based on the distribution of the simulated model. We also give an overview of all the notations that are used in a nested simulation.

It is usually not the case that the loss distribution of a portfolio is fully available. However, a computer program can simulate the returns of an investment under different scenarios based on a financial model. These simulated returns are used to analyze the risks of an investment. Simulating and analyzing risks is almost always done with the Monte Carlo method. This means that it is simulated many times, and the result is a distribution function representing the range of the possible outcomes of the returns. Risk measures often only take a small part of the data involving the lowest returns into account. Therefore, a large number of observations are required to obtain an accurate estimate. A lot of research has focussed in recent decades on applying the Monte Carlo methods for the approximation of risk measures. Their goal is to make an accurate approximation with a fixed amount of computational costs.

3.1. Nested simulation

The focus of our thesis is on a nested simulation framework. This means that we can divide the simulation into two stages. The first stage samples different risk scenarios independently, this is called the outer stage. The second stage samples the corresponding return of the portfolio at a given time horizon, conditional on a risk scenario. This stage is called the inner stage. Let $\theta : \Omega \to \Theta$ be a random variable that describes the risk scenario. Let $X(\theta) : \Theta \to \mathbb{R}$ be a function that describes the return of the portfolio for a given time horizon. We define $X^{\theta} = X(\theta)$. The space of possible risk scenarios Ω, is equipped with the probability measure $\mathbb{P}$. The probability measure $\mathbb{P}$ is known as the real world or physical measure.

Let $(\theta(\omega_m))_{m\geq 1}$ be a sequence of risk scenarios that consists of independent and identically distributed (i.i.d.) random variables. These risk scenarios are generated according to the real world distribution Ω. The portfolio return $X^{\theta(\omega)}$ is dependent on the value $\theta(\omega) \in \Theta$. We assume that the expression $X^{\theta(\omega)}$ is too complex to calculate analytically, for a given risk scenario $\theta(\omega)$, because $X^{\theta(\omega)}$ can represent the valuation of an entire portfolio with potentially complex financial products. Therefore, we assume that there are parts of the portfolio that cannot be valued analytically and that only Monte Carlo simulations are feasible for the valuation of the portfolio. So this means that the value of the portfolio return $X^{\theta(\omega)}$ is already fixed for a given value of $\theta(\omega)$ and is therefore not a variable, but since it is too complex to compute, the value $X^{\theta(\omega)}$ is estimated through Monte Carlo simulations. Let $\hat{X}^{\theta(\omega)} : \Omega_2 \to \mathbb{R}$, be a random variable where holds that $\mathbb{E}\left[\hat{X}^{\theta(\omega)}\right] = X^{\theta(\omega)}$, for a fixed risk scenario $\theta(\omega) \in \Theta$. The probability space Ω_2 is equipped with the probability measure $\mathbb{Q}$, where $\mathbb{Q}$ is defined as the risk-neutral measure. We use $\hat{X}^{\theta(\omega)}$ as an estimator for $X^{\theta(\omega)}$.

Suppose that the outer stage generates a sequence of M risk scenarios $(\theta(\omega_i))_{i=1}^{M}$. The approximations for

$X^{\theta(\omega_i)}$, where $i \in \{1, ..., M\}$, are made in the inner stage of the simulation. For a given value of $\theta(\omega_i)$, is an inner stage sample denoted as $\hat{X}_{j:N}^{\theta(\omega_i)}$. The inner stage sample $\hat{X}_{j:N}^{\theta(\omega_i)}$ is just one of the N samples that are generated for the given risk scenario $\theta(\omega_i)$, where $j \in \{1, ..., N\}$. The sequence $\left(\hat{X}_{j:N}^{\theta(\omega_i)}\right)_{j=1}^{N}$ consists of i.i.d. random variables that are generated according to the risk neutral distribution Ω_2. Figure 3.1 illustrates how this nested simulation works.

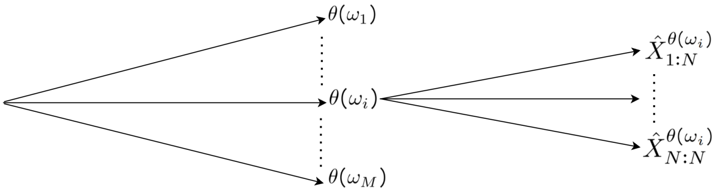

$\theta(\omega_1)$
$\theta(\omega_i)$
$\theta(\omega_M)$
$\hat{X}_{1:N}^{\theta(\omega_i)}$
$\hat{X}_{N:N}^{\theta(\omega_i)}$

Figure 3.1: The outer stage generates the sequence $(\theta(\omega_m))_{m=1}^{M}$ of i.i.d. risk scenarios. Conditional on each risk scenario $\theta(\omega_i)$ for $i \in \{1, ..., M\}$, generates the inner stage a number of N i.i.d. samples $\hat{X}_{j:N}^{\theta(\omega_i)}$, where $j \in \{1, ..., N\}$.

These inner stage samples are used to estimate the portfolio return. So suppose that we want to estimate $X^{\theta(\omega_i)}$ and we have the sequence of generated inner stage samples $\left(\hat{X}_{j:N}^{\theta(\omega_i)}\right)_{j=1}^{N}$, then the estimator for $X^{\theta(\omega_i)}$ is computed as follows,

$$\hat{X}_{N}^{\theta(\omega_i)} = \frac{1}{N} \sum_{j=1}^{N} \hat{X}_{j:N}^{\theta(\omega_i)}. \tag{3.1}$$

Here is $\hat{X}_{N}^{\theta(\omega_i)}$ the estimator for $X^{\theta(\omega_i)}$, which is computed by N different inner stage samples.

3.2. Approximation of the expected shortfall

We want to approximate the expected shortfall for a given time horizon. The expected shortfall is defined in equation (2.8), where X represents the return of a portfolio. For the approximation of the expected shortfall, we generate the sequence $(\theta(\omega_m))_{m\geq 1}$ of i.i.d. samples that represent the risk scenarios. Conditional on each of those risk scenarios $\theta(\omega_i) \in (\theta(\omega_m))_{m\geq 1}$, we evaluate a possible return of the portfolio $\hat{X}^{\theta(\omega_i)}$. This gives us a distribution of the random variable X^{θ}, which represents the possible returns of the portfolio for a given time horizon. To determine the expected shortfall, we need to acquire the distribution of X^{θ}. We are not interested in the expectation $\mathbb{E}_{\mathbb{P}\otimes\mathbb{Q}}\left[X^{\theta}\right]$.

Now we want to set up a formula that allows us to compute the expected shortfall for a given simulated distribution of X^{θ}. Acerbi [2] showed, with the help of the definition of the expected shortfall in (2.8) and the definition of the quantiles in (2.5) and (2.6), that the expected shortfall can also be defined as follows,

$$\mathrm{ES}_{\alpha}(X) = -\frac{1}{\alpha}\left(\mathbb{E}[X\mathbf{1}_{\{X\leq s\}}] + s(\alpha - \mathbb{P}(X \leq s))\right), \tag{3.2}$$

with $s \in \left[x_{(\alpha)}, x^{(\alpha)}\right]$. Proposition 3.2.1 shows how a sampled sequence can be used to approximate the expected shortfall. The definition in (3.2) is used to prove proposition 3.2.1.

Proposition 3.2.1 Let X be a random variable with $\mathbf{E}[X^{-}] < \infty$ and $(X^1, X^2, ..., X^M)$ be an independent sequence of random variables with the same distribution as X. Denote $X^{1:M} \leq X^{2:M} \leq \cdots \leq X^{M:M}$ as the ordered components of the independent sequence $(X^1, X^2, ..., X^M)$. Then, for a fixed $\alpha \in (0, 1)$ we have that the following holds,

$$-\lim_{M\to\infty} \frac{\sum_{i=1}^{\lfloor M\alpha \rfloor} X^{i:M}}{\lfloor M\alpha \rfloor} = \mathrm{ES}_{\alpha}(X) \tag{3.3}$$

where $\lfloor x \rfloor$ denotes the largest integer smaller or equal to x.

Proof. We have that:

$$
\begin{aligned}
\frac{\sum_{i=1}^{\lfloor M\alpha\rfloor} X^{i:M}}{\lfloor M\alpha\rfloor} &= \frac{\sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\{1,\dots,\lfloor M\alpha\rfloor\}(i)}}{\lfloor M\alpha\rfloor} \\
&= \frac{1}{\lfloor M\alpha\rfloor}\left(\sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\{1,\dots,\lfloor M\alpha\rfloor\}}(i) + \sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}} - \sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}}\right) \\
&= \frac{1}{\lfloor M\alpha\rfloor}\left(\sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}} + \sum_{i=1}^{M} X^{i:M}\left(\mathbf{1}_{\{1,\dots,\lfloor M\alpha\rfloor\}}(i) - \mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}}\right)\right) \\
&= \frac{1}{\lfloor M\alpha\rfloor}\left(\sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}} + X^{\lfloor M\alpha\rfloor:M}\sum_{i=1}^{M}\left(\mathbf{1}_{\{1,\dots,\lfloor M\alpha\rfloor\}}(i) - \mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}}\right)\right) \\
&= \frac{1}{\lfloor M\alpha\rfloor}\left(\sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}} + X^{\lfloor M\alpha\rfloor:M}\left(\lfloor M\alpha\rfloor - \sum_{i=1}^{M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}}\right)\right) \\
&= \frac{M}{\lfloor M\alpha\rfloor}\left(\frac{1}{M}\sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}} + X^{\lfloor M\alpha\rfloor:M}\left(\frac{\lfloor M\alpha\rfloor}{M} - \frac{1}{M}\sum_{i=1}^{M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}}\right)\right). \qquad (3.4)
\end{aligned}
$$

We have that the following limits hold:

$$
\lim_{M\to\infty}\inf X^{\lfloor M\alpha\rfloor:M} = x_{(\alpha)}, \qquad \lim_{M\to\infty}\sup X^{\lfloor M\alpha\rfloor:M} = x^{(\alpha)}, \qquad \lim_{M\to\infty}\frac{\lfloor M\alpha\rfloor}{M} = \alpha \qquad (3.5)
$$

By using these limits and the Glivenko-Cantelli theorem, we get for $s\in\left[x_{(\alpha)}, x^{(\alpha)}\right]$

$$
\begin{aligned}
\lim_{M\to\infty}\frac{\sum_{i=1}^{\lfloor M\alpha\rfloor} X^{i:M}}{\lfloor M\alpha\rfloor} &= \lim_{M\to\infty}\frac{M}{\lfloor M\alpha\rfloor}\left(\frac{1}{M}\sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}} + X^{\lfloor M\alpha\rfloor:M}\left(\frac{\lfloor M\alpha\rfloor}{M} - \frac{1}{M}\sum_{i=1}^{M}\mathbf{1}_{\left\{X^{i:M}\le X^{\lfloor M\alpha\rfloor:M}\right\}}\right)\right) \\
&= \lim_{M\to\infty}\frac{1}{\alpha}\left(\frac{1}{M}\sum_{i=1}^{M} X^{i:M}\mathbf{1}_{\{X^{i:M}\le s\}} + s\left(\alpha - \frac{1}{M}\sum_{i=1}^{M}\mathbf{1}_{\{X^{i:M}\le s\}}\right)\right) \\
&= \frac{1}{\alpha}\left(\mathbb{E}[X\mathbf{1}_{\{X\le s\}}] + s(\alpha - \mathbb{P}(X\le s))\right) \qquad (3.6)
\end{aligned}
$$

With the help of equation (3.2), we have that the equality in (3.3) holds. □

Our thesis denotes an estimate of the expected shortfall as $\hat{c}_\alpha$, and its real value is denoted as c_α.

3.3. Problem stetting

In our numerical experiments, we focus on estimating the expected shortfall of a financial portfolio that consists of one European call option. The underlying stock price is modelled as a Stochastic Differential Equation (SDE). The outer stages are simulated over the interval $[0,\tau]$ and the inner stages over the interval $[\tau, T]$. We have the following stock price process for $t\in[0,\tau]$,

$$
\begin{cases}
\mathrm{d}S(t) &= \mu S(t)\mathrm{d}t + \sigma S(t)\mathrm{d}\omega(t) \\
S(0) &= S_0
\end{cases} \qquad (3.7)
$$

here is $S(t)$ the stochastic process of the stock price, μ the constant deterministic growth rate of the stock, σ the constant percentage volatility parameter and $\omega(t)$ a Wiener process. The inner stages are simulated over the interval $[\tau, T]$ and have the following stock price process,

$$
\begin{cases}
\mathrm{d}S(t) &= rS(t)\mathrm{d}t + \sigma S(t)\mathrm{d}W(t) \\
S(\tau) &= S_\tau
\end{cases} \qquad (3.8)
$$

here is r the so-called risk-free interest rate and $W(t)$ also a Wiener process. The reason that we picked two different stock price processes on the interval $[0,T]$, is because the outer stages are generated according to the real world distribution governed by the drift μ, whereas the inner stages are generated according to the risk neutral distribution governed by the drift r. Such SDE's are called Geometric Brownian Motions (GBM), and its analytical solution on the interval $t \in [0,T]$ is given by,

$$S(t) = \begin{cases} S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma\omega(t)\right), & \text{for } t \in [0,\tau] \\ S_\tau \exp\left(\left(r - \frac{1}{2}\sigma^2\right)(t-\tau) + \sigma W(t-\tau)\right), & \text{for } t \in [\tau,T] \end{cases}. \tag{3.9}$$

Some SDE's are not always solvable analytically or hard to solve. In finance, modelling a stock price is one of the most commonly studied problems, where the dynamics of these stock prices are usually modelled as an SDE. We picked GMB's because the solution can be derived analytically, which means that we can check how accurate the approximation of the expected shortfall is. We focus in our numerical experiment on the value of a European call option, where the portfolio return is equal to the price of the European call option. Therefore, the return of the portfolio at time τ is equal to,

$$X^{\theta(\omega)} = \mathbb{E}_{\mathbb{Q}}\left[e^{-r(T-\tau)} \max(S_T(\omega,W) - K, 0) \middle| \theta(\omega)\right] \tag{3.10}$$

where

$$S_T(\omega,W) = S_\tau(\omega)e^{(r-\sigma^2/2)(T-\tau)+\sigma\sqrt{T-\tau}W} \quad \text{and} \quad S_\tau(\omega) = S_0 e^{(\mu-\sigma^2/2)\tau+\sigma\sqrt{\tau}\omega}. \tag{3.11}$$

We generate an outer stage sample according to $\theta(\omega_i) = S_\tau(\omega_i)$, where ω_i is a standard random normal variable. So the outer stage sample represents the stock price at time τ. The variable $S_T(\omega_i, W_{i,j})$ is the stock price at time T and is dependent on the value $S_\tau(\omega_i)$, here is $W_{i,j}$ also a standard random normal variable. We compute an inner stage sample with the help of $S_T(\omega_i, W_{i,j})$. Figure 3.2 illustrates an example of how the stock prices are generated. In this example we generate a total of M i.i.d. outer stage samples, and for each outer stage sample we generate N i.i.d. stock prices for time T.

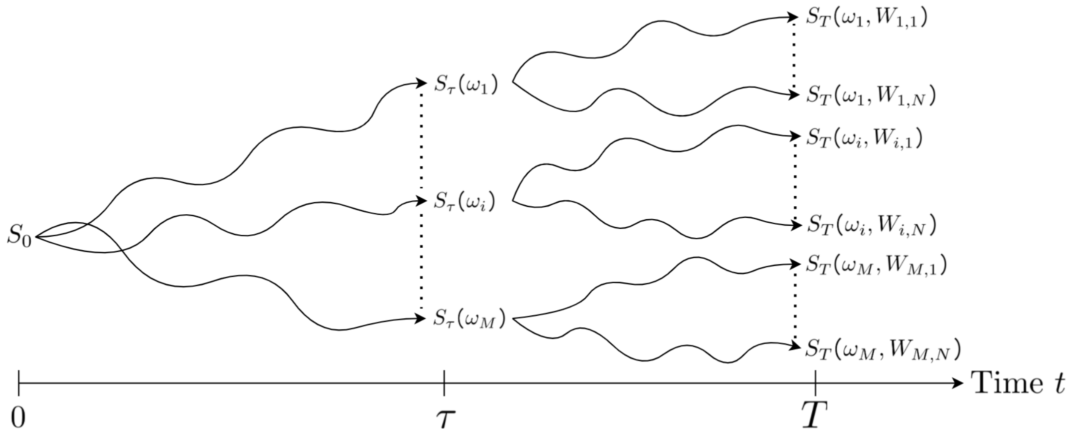


Figure 3.2: Display of an example where the outer stage generates M i.i.d. stock prices for time τ. Conditional on each of these stock prices at time τ, the inner stage generates N i.i.d. stock prices for time T.

An inner stage sample is then computed as follows with the help of a generated stock price $S_T(\omega_i, W_{i,j})$,

$$\hat{X}_{j:N}^{\theta(\omega_i)} = e^{-r(T-\tau)} \max\left(S_T(\omega_i, W_{i,j}) - K, 0\right), \tag{3.12}$$

for all $i \in \{1,..,M\}$ and for all $j \in \{1,..,N\}$. Table 3.1 gives an overview of the notations.

θ	A random variable that represents the risk scenario at the risk time horizon
$\theta(\omega)$	A generated outer stage sample that represents a risk scenario at the risk time horizon
X^{θ}	Portfolio return at risk time horizon under risk scenario θ
$X^{\theta(\omega)}$	Portfolio return at risk time horizon under risk scenario $\theta(\omega)$
$\hat{X}_{j:N}^{\theta(\omega)}$	The j-th generated inner stage sample that represents a possible return of the portfolio under risk scenario $\theta(\omega)$, where $j \in \{1,...,N\}$
$\hat{X}_{N}^{\theta(\omega)}$	An estimator for the portfolio return $X^{\theta(\omega)}$, that is computed by N i.i.d. inner stage samples
c_α	The expected shortfall of X^{θ}
$\hat{c}_\alpha$	An estimator of the expected shortfall c_α
τ	The risk time horizon
T	The final time horizon
$S(t)$	The stochastic process of the stock price, where $t \in [0,T]$
μ	The constant deterministic growth of the stock price
σ	The constant volatility of the stock price
r	The constant risk-free interest rate of the stock price

Table 3.1: Notations

3.3.1. Deriving the expected shortfall analytically

We want to determine what the value of the expected shortfall is, because then we can examine how accurate an estimate is of the expected shortfall. First, we have to determine the value of the portfolio return $X^{\theta(\omega)}$ for a given risk scenario $\theta(\omega)$. The European call option price can be derived analytically with the use of the Black-Scholes formula. Black-Scholes show us that the return of the portfolio in (3.10) at time τ is equal to,

$$V_c(S_\tau) = S_\tau F_{\mathcal{N}(0,1)}(d_1) - Ke^{-r(T-\tau)} F_{\mathcal{N}(0,1)}(d_2) \tag{3.13}$$

with

$$d_1 = \frac{\log \frac{S_\tau}{K} + \left(r + \sigma^2/2\right)(T-\tau)}{\sigma\sqrt{T-\tau}}, \tag{3.14}$$

$$d_2 = \frac{\log \frac{S_\tau}{K} + \left(r - \sigma^2/2\right)(T-\tau)}{\sigma\sqrt{T-\tau}}, \tag{3.15}$$

where $F_{\mathcal{N}(0,1)}(\cdot)$ is the cumulative distribution function (CDF) of a standard normal variable. Therefore, for a given outer stage sample $\theta(\omega_i) = S_\tau(\omega_i)$, the portfolio return $X^{\theta(\omega_i)}$ is equal to $V_c(S_\tau(\omega_i))$. The distribution of X^{θ} is therefore equal to $V_c(S_\tau)$. We use definition (2.22) to compute the expected shortfall. Here is $q_\alpha(X)$ defined as $\inf\{x \in \mathbb{R} : \mathbb{P}(X \le x) \ge \alpha\}$. We can derive $\mathbb{P}(X^{\theta} \le x)$ as follows, with the help of the Black-Scholes formula in (3.13) and the analytical solution of the GBM on the interval $[0,\tau]$ in (3.9),

$$\begin{aligned}
\mathbb{P}\left(X^{\theta} \le x\right) &= \mathbb{P}\left(V_c(S_\tau) \le x\right) \\
&= \mathbb{P}\left(V_c\left(S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma\sqrt{\tau}\omega\right)\right) \le x\right) \\
&= F_{\mathcal{N}(0,1)}\left(\frac{\log \frac{V_c^{-1}(x)}{S_0} - \left(\mu - \sigma^2/2\right)\tau}{\sigma\sqrt{\tau}}\right).
\end{aligned} \tag{3.16}$$

Next we have that,

$$\begin{aligned}
\mathbb{P}(X^{\theta} \le x) \ge \alpha \quad &\Longrightarrow \quad F_{\mathcal{N}(0,1)}\left(\frac{\log \frac{V_c^{-1}(x)}{S_0} - \left(\mu - \sigma^2/2\right)\tau}{\sigma\sqrt{\tau}}\right) \ge \alpha \\
&\Longrightarrow \quad x \ge V_c\left(S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma F_{\mathcal{N}(0,1)}^{-1}(\alpha)\right)\right).
\end{aligned} \tag{3.17}$$

Finally, equation (2.22) shows us that the expected shortfall can be computed as follows,

$$c_\alpha = \mathrm{ES}_\alpha\left(X^\theta\right) = -\frac{1}{\alpha}\int_0^\alpha V_c(S_\tau(\omega_y))\,\mathrm{d}y \tag{3.18}$$

with,

$$S_\tau(\omega_y) = S_0 e^{(\mu-\sigma^2/2)\tau+\sigma\sqrt{\tau}\omega_y} \quad \text{and} \quad \omega_y = F^{-1}_{\mathcal{N}(0,1)}(y). \tag{3.19}$$

4
Standard Monte Carlo method

In this section, we discuss how the standard Monte Carlo method operates. First, we describe how the simulated returns in the standard Monte Carlo method are determined. Next, we examine how to apply this method in a nested simulation optimally. Finally, we give an explicit numerical algorithm that uses the correct input parameters to illustrate how this method operates.

4.1. The standard Monte Carlo method applied in a nested simulation

Nested simulations can be divided into two stages, such as described in section 3.1. In the standard Monte Carlo method we sample uniformly, this means that we use the same number of inner stage samples for each risk scenario in the simulation. So the number of inner stage samples does not depend on the risk scenario $\theta(\omega)$. We denote the number M as the total number of sampled risk scenarios in the standard Monte Carlo method. Conditional on each of these M risk scenarios, we generate N potential portfolio returns at a given time horizon. So the number N is denoted as the total number of inner stage samples. The total computational cost of the standard Monte Carlo method is thus equal to $M \cdot N$, since for each outer stage sample, generates the standard Monte Carlo method N corresponding inner stage samples.

Suppose that the standard Monte Carlo method generates a sequence $(\theta(\omega_i))_{i=1}^{M}$ of i.i.d. risk scenarios. Conditional on each risk scenario $\theta(\omega_i)$ for $i \in \{1,...,M\}$, generates the inner stage a number of N i.i.d. samples $\hat{X}_{j:N}^{\theta(\omega_i)}$, where $j \in \{1,...,N\}$. With the help of equation (3.1), do we obtain the sequence $\left(\hat{X}_N^{\theta(\omega_1)}, \hat{X}_N^{\theta(\omega_2)}, ..., \hat{X}_N^{\theta(\omega_M)}\right)$ of simulated returns, each computed by N inner stage samples. The simulated returns are used to approximate the expected shortfall with the help of proposition 3.2.1. First, we have to put the simulated returns in increasing order. We apply the permutation $\mathbf{m}$ on $[\![\theta(\omega_1), \theta(\omega_M)]\!]$, such that the following holds,

$$\hat{X}_N^{\mathbf{m}(\theta(\omega_1))} \leq \hat{X}_N^{\mathbf{m}(\theta(\omega_2))} \leq ... \leq \hat{X}_N^{\mathbf{m}(\theta(\omega_M))}. \tag{4.1}$$

From proposition 3.2.1 follows that the expected shortfall can be approximated as,

$$\hat{c}_\alpha = -\frac{1}{\lfloor M\alpha \rfloor} \sum_{i=1}^{\lfloor M\alpha \rfloor} \hat{X}_N^{\mathbf{m}(\theta(\omega_i))}. \tag{4.2}$$

The accuracy of the estimator $\hat{c}_\alpha$ depends a lot on what the ratio is between the values of M and N. Gordy and Juneja [30] studied what the optimal ratio is between the inner and outer stage samples in a nested simulation for the approximation of the expected shortfall.

4.2. Optimally allocating the computational cost between the inner and the outer stage

The estimator's accuracy for c_α, is dependent on the two parameters M and N. We have to determine what the best ratio is between these parameters. The Mean Squared Error (MSE) is a helpful tool to establish if an estimate is accurate. Therefore, can we find the optimal ratio for M and N when we minimize the MSE of an estimator with a limited computational cost for the simulation. Let C be defined as the computational cost for the simulation, then we obtain the following optimization problem:

$$\begin{aligned} \underset{M\ N}{\text{minimize}} \quad & \mathbb{E}\left[\left(\hat{c}_\alpha - c_\alpha\right)^2\right] \\ \text{subject to} \quad & MN \leq C \\ & M, N \in \mathbb{N} \end{aligned} \tag{4.3}$$

This optimization problem returns the values of M and N such that the MSE is minimized. To simplify the optimization problem, we split the MSE into two parts as follows:

$$\begin{aligned} \mathbb{E}\left[\left(\hat{c}_\alpha - c_\alpha\right)^2\right] &= \mathbb{E}\left[\left(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha] + \mathbb{E}[\hat{c}_\alpha] - c_\alpha\right)^2\right] \\ &= \mathbb{E}\left[\left(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha]\right)^2 + 2\left(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha]\right)\left(\mathbb{E}[\hat{c}_\alpha] - c_\alpha\right) + \left(\mathbb{E}[\hat{c}_\alpha] - c_\alpha\right)^2\right] \\ &= \mathbb{E}\left[\left(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha]\right)^2\right] + \mathbb{E}\left[2\left(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha]\right)\left(\mathbb{E}[\hat{c}_\alpha] - c_\alpha\right)\right] + \mathbb{E}\left[\left(\mathbb{E}[\hat{c}_\alpha] - c_\alpha\right)^2\right] \\ &= \mathbb{E}\left[\left(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha]\right)^2\right] + 2\mathbb{E}\left[\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha]\right]\left(\mathbb{E}[\hat{c}_\alpha] - c_\alpha\right) + \left(\mathbb{E}[\hat{c}_\alpha] - c_\alpha\right)^2 \\ &= \mathbb{E}\left[\left(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha]\right)^2\right] + 2\left(\mathbb{E}[\hat{c}_\alpha] - \mathbb{E}[\hat{c}_\alpha]\right)\left(\mathbb{E}[\hat{c}_\alpha] - c_\alpha\right) + \left(\mathbb{E}[\hat{c}_\alpha] - c_\alpha\right)^2 \\ &= \underbrace{\mathbb{E}\left[\left(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha]\right)^2\right]}_{\text{variance}} + \underbrace{\left(\mathbb{E}[\hat{c}_\alpha - c_\alpha]\right)^2}_{\text{bias}^2} \end{aligned} \tag{4.4}$$

The following mild assumption is applied to solve the optimization problem in (4.3):

Assumption 4.2.1 Denote $X^{\theta(\omega)}$ the portfolio value under risk scenario $\theta(\omega)$, and denote by $\hat{X}_N^{\theta(\omega)}$ an estimator of the form (3.1) for $X^{\theta(\omega)}$, that is based on the average of N i.i.d. inner stage samples. Assume the following

1. The joint probability density function $g_N(\cdot,\cdot)$ of $\left(X^\theta, \hat{X}_N^\theta\right)$ and its partial derivatives $(\partial/\partial x)g_N(x,\hat{x})$ and $(\partial^2/\partial x^2)g_N(x,\hat{x})$ exists for each N and $(x,\hat{x})$.

2. For each $N \geq 1$ there exist functions $p_{0,N}(\cdot)$, $p_{1,N}(\cdot)$ and $p_{2,N}(\cdot)$ such that

$$g_N(x,\hat{x}) \leq p_{0,N}(\hat{x}), \qquad \left|\frac{\partial}{\partial x} g_N(x,\hat{x})\right| \leq p_{1,N}(\hat{x}), \qquad \left|\frac{\partial^2}{\partial x^2} g_N(x,\hat{x})\right| \leq p_{2,N}(\hat{x}), \tag{4.5}$$

for all $(x,\hat{x})$. Furthermore,

$$\sup_N \int_{-\infty}^{\infty} |\hat{x}|^r\, p_{i,N}(\hat{x})\mathrm{d}\hat{x} < \infty \tag{4.6}$$

for all $i = 0,1,2$, and $0 \leq r \leq 4$.

Proposition 4.2.2 is established by Gordy and Juneja [30], under the condition that assumption 4.2.1 holds.

Proposition 4.2.2 Suppose that assumption 4.2.1 holds, as $N \to \infty$, then satisfies the estimator's variance and bias that

$$\mathbb{E}\left[\left(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha]\right)^2\right] = \frac{\mathbb{V}[X^\theta \mathbf{1}_{\{X^\theta \leq x_{(\alpha)}\}}] + \mathcal{O}_N(1/N)}{\alpha^2 M} \tag{4.7}$$

and

$$\left(\mathbb{E}[\hat{c}_\alpha - c_\alpha]\right)^2 = \frac{\left(\Upsilon(x_{(\alpha)}) - x_{(\alpha)}\Upsilon'(x_{(\alpha)})\right)^2}{\alpha^2 N^2} + \mathcal{O}_N(1/N^{5/2}) \tag{4.8}$$

with $\Upsilon(u) = \frac{1}{2}f(u)\mathbb{E}\left[\sigma(\theta(\omega_i))\middle|X^{\theta(\omega_i)} = -u\right]$ and where $f(\cdot)$ is the density function of the variable X^θ and $\sigma(\theta(\omega_i))$ is the variance of the inner stage samples under risk scenario $\theta(\omega_i)$.

Now we obtain the following optimization problem after we substitute (4.4), (4.7), and (4.8) in (4.3):

$$\begin{array}{ll} \underset{M\ N}{\text{minimize}} & \dfrac{\mathbb{V}[X^{\theta}\mathbf{1}_{\{X^{\theta}\leq x_{(\alpha)}\}}]+\mathcal{O}_N(1/N)}{\alpha^2 M}+\dfrac{\left(\Upsilon(x_{(\alpha)})-x_{(\alpha)}\Upsilon'(x_{(\alpha)})\right)^2}{\alpha^2 N^2}+\mathcal{O}_N(1/N^{5/2}) \\ \text{subject to} & MN\leq C \\ & M,N\in\mathbb{N} \end{array} \tag{4.9}$$

We apply the Lagrange multiplier method for finding the optimal value for M and N. The Lagrange multiplier method works as follows, to find the minimum of a function $f(x)$ under the constraint that $g(x)=0$, is the Lagrangian function formed as follows,

$$\mathcal{L}(x,\lambda)=f(x)-\lambda g(x). \tag{4.10}$$

The solution of the optimization problem is the saddle point of $\mathcal{L}$ as a function of x and λ. We want to minimize the MSE as a function of M and N. The constraint function in the Lagrange multiplier method must also contain M and N, so we obtain the following Lagrangian function,

$$\mathcal{L}(M,N,\lambda)=\frac{\gamma_1}{M}+\frac{\gamma_2}{N^2}+\lambda(MN-C), \tag{4.11}$$

where

$$\gamma_1=\frac{\mathbb{V}[X^{\theta}\mathbf{1}_{\{X^{\theta}\leq x_{(\alpha)}\}}]}{\alpha^2} \quad \text{and} \quad \gamma_2=\frac{\left(\Upsilon(x_{(\alpha)})-x_{(\alpha)}\Upsilon'(x_{(\alpha)})\right)^2}{\alpha^2}. \tag{4.12}$$

Now we are going to find the stationary points of $\mathcal{L}(M,N,\lambda)$:

$$\frac{\delta\mathcal{L}}{\delta M}=0 \iff -\frac{\gamma_1}{M^2}-\lambda N=0 \iff \lambda=-\frac{\gamma_1}{M^2N} \tag{4.13}$$

$$\frac{\delta\mathcal{L}}{\delta N}=0 \iff -\frac{2\gamma_2}{N^3}-\lambda M=0 \iff -\frac{2\gamma_2}{N^3}+\frac{\gamma_1}{MN}=0 \iff M=\frac{\gamma_1 N^2}{2\gamma_2} \tag{4.14}$$

$$\frac{\delta\mathcal{L}}{\delta\lambda}=0 \iff MN-C=0 \iff \frac{\gamma_1 N^3}{2\gamma_1}=C \tag{4.15}$$

Therefore, the optimal values for N and M are denoted as follows:

$$N^*=\left(\frac{2\gamma_2}{\gamma_1}\right)^{1/3}C^{1/3}, \quad M^*=\left(\frac{\gamma_1}{2\gamma_2}\right)^{1/3}C^{2/3} \tag{4.16}$$

4.3. Numerical algorithm standard Monte Carlo method

In this work, we use the standard Monte Carlo method in a simulation as follows for the given values of M, N and α:

Algorithm 1 standard Monte Carlo method

```
procedure StandardMC($M, N, \alpha$)
    for $i \leftarrow 1$ to $M$ do
        generate risk scenario $\theta(\omega_i)$
        for $j \leftarrow 1$ to $N$ do
            conditional on risk scenario $\theta(\omega_i)$, generate return $\hat{X}_{j:N}^{\theta(\omega_i)}$
        end for
        $\hat{X}_N^{\theta(\omega_i)} \leftarrow \frac{1}{N}\sum_{j=1}^{N}\hat{X}_{j:N}^{\theta(\omega_i)}$
    end for
    use permutation $\mathbf{m}$ on $[\![\theta(\omega_1),\theta(\omega_M)]\!]$ to put the simulated returns in an increasing order,

        $\hat{X}_N^{\mathbf{m}(\theta(\omega_1))} \leq \hat{X}_N^{\mathbf{m}(\theta(\omega_2))} \leq \cdots \leq \hat{X}_N^{\mathbf{m}(\theta(\omega_M))}$

    $\hat{c}_\alpha \leftarrow \frac{1}{\lfloor M\alpha\rfloor}\sum_{i=1}^{\lfloor M\alpha\rfloor}\hat{X}_N^{\mathbf{m}(\theta(\omega_i))}$
    return $\hat{c}_\alpha$
end procedure
```

5
Adaptive sampling

In this section, we discuss why the adaptive sampling technique can be efficient when applied to a nested simulation. We also explain how adaptive sampling can be applied optimally to estimate the expected shortfall. In addition, we provide a numerical algorithm that shows how adaptive sampling operates, given the correct input parameters.

5.1. The idea behind adaptive sampling

The standard Monte Carlo method that is described in section 4, employs a constant number of N inner stage samples for the approximation of $\hat{X}_N^{\theta(\omega_i)}$, for each $i \in \{1,...,M\}$. A nested simulation can become more efficient by distributing the computational cost non-uniformly over the different risk scenarios. For each risk scenario the estimated return of the portfolio is not of equal importance. It is more efficient to adjust the number of inner stage samples accordingly. The goal at the end is to make an accurate approximation of the expected shortfall. This risk measure is computed by the lowest values in the distribution of the possible returns of the portfolio, therefore is it important that these values have a high accuracy. A good example of this is shown in Figure 5.1. Suppose that we have a distribution of the portfolio return X^θ, which is illustrated by the black line. Let then $\theta(\omega_1)$ and $\theta(\omega_2)$ be two generated risk scenarios. The corresponding portfolio returns are approximated in the inner stage of the nested simulation. Let the dotted blue lines display the distribution of the inner stage samples $\left(\hat{X}_{j:N_1}^{\theta(\omega_1)}\right)_{j=1}^{N_1}$ and $\left(\hat{X}_{j:N_2}^{\theta(\omega_2)}\right)_{j=1}^{N_2}$. As already explained, are these inner stage samples used to compute the estimators $\hat{X}_{N_1}^{\theta(\omega_1)}$ and $\hat{X}_{N_2}^{\theta(\omega_2)}$. The portfolio returns in the red area contribute to the computation of the expected shortfall.

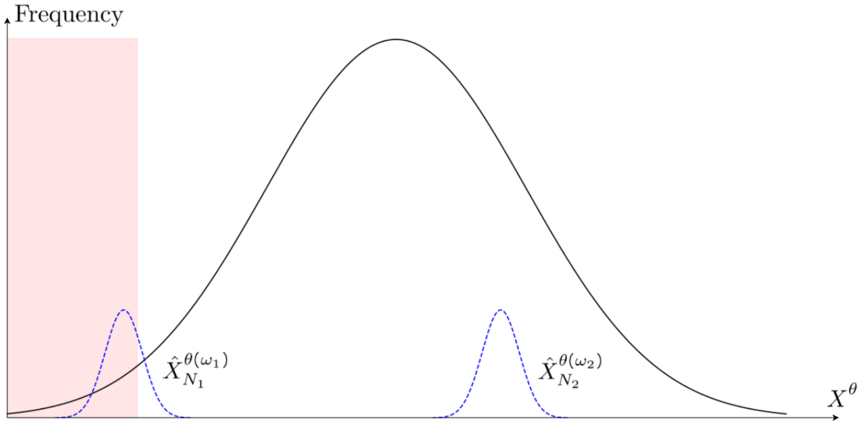

Frequency
$\hat{X}_{N_1}^{\theta(\omega_1)}$
$\hat{X}_{N_2}^{\theta(\omega_2)}$
X^θ

Figure 5.1: Example of the advantages of adaptive sampling. The estimator $\hat{X}_{N_2}^{\theta(\omega_2)}$ with the corresponding risk scenario $\theta(\omega_2)$ is unlikely to contribute to the computation of the expected shortfall. Therefore, the number of inner stage samples N_2 can be set small. The estimator $\hat{X}_{N_1}^{\theta(\omega_1)}$ is likely to contribute to the computation of the expected shortfall. Hence, the number of inner stage samples N_1 in this scenario should be set large.

Figure 5.1 displays the distribution of the estimators $\left(\hat{X}_{j:N_1}^{\theta(\omega_1)}\right)_{j=1}^{N_1}$ and $\left(\hat{X}_{j:N_2}^{\theta(\omega_2)}\right)_{j=1}^{N_2}$, for the corresponding risk scenarios $\theta(\omega_1)$ and $\theta(\omega_2)$. It is almost certain that the value for $\hat{X}_{N_2}^{\theta(\omega_2)}$ is not in the red area, so it does not contribute to the computation of the expected shortfall. While the largest part of the distribution $\left(\hat{X}_{j:N_1}^{\theta(\omega_1)}\right)_{j=1}^{N_1}$ is in the red area, therefore it is likely that $\hat{X}_{N_1}^{\theta(\omega_1)}$ shall contribute to the computation of the expected shortfall. Thus, it would make more sense to use more inner stage samples for the estimator $\hat{X}_{N_1}^{\theta(\omega_1)}$, in comparison with the estimator $\hat{X}_{N_2}^{\theta(\omega_2)}$. Since more inner stage samples for an estimator results in a more accurate estimate.

5.2. Adaptive sampling by detection of the lowest estimators

The algorithm where we will be focussing on is based on the work of Bouchard, Reghai, and Virrion [6]. They apply adaptive sampling in a nested simulation for the computation of the expected shortfall. The basic idea of their multistep algorithm is that the estimators who contribute to the computation of the expected shortfall have higher accuracy than the estimators that do not. So for a given risk scenario $\theta(\omega_i)$, where $i \in \{1,...,M\}$, the estimator for $X^{\theta(\omega_i)}$ has more inner stage samples added if its value is in the lowest tail of the distribution. The only problem is that it is unknown beforehand of the simulation which risk scenarios contribute to the computation of the expected shortfall. The algorithm solves this problem by first generating the estimators for $X^{\theta(\omega_i)}$ with a small number of inner stage samples, for each $i \in \{1,...,M\}$. These simulated returns are ranked, and the lowest $q_1 \leq M$ are selected. The selected returns are approximated again by adding more inner stage samples to their estimators. Then we rank the estimators for $X^{\theta(\omega_i)}$ again, for each $i \in \{1,...,q_1\}$. The lowest q_2 returns, where $q_2 \leq q_1$, are selected and so on. After $L \geq 2$ steps, we compute the expected shortfall with the remaining simulated returns. An example is stated below of how this algorithm operates when approximating the expected shortfall.

Just as with the standard Monte Carlo method, we start by generating a number of M i.i.d. samples that represent the risk scenarios, which are denoted as $(\theta(\omega_1),...,\theta(\omega_M))$. Next, we approximate the corresponding returns $\left(\hat{X}_{N_1}^{\theta(\omega_1)},...,\hat{X}_{N_1}^{\theta(\omega_M)}\right)$, for each risk scenario. We use for each simulated return N_1 inner stage samples, for some $N_1 \geq 1$. The permutation $\mathbf{m}_1$ on $[\![\theta(\omega_1),\theta(\omega_M)]\!]$, puts the simulated returns in an increasing order,

$$\hat{X}_{N_1}^{\mathbf{m}_1(\theta(\omega_1))} \leq \hat{X}_{N_1}^{\mathbf{m}_1(\theta(\omega_2))} \leq ... \leq \hat{X}_{N_1}^{\mathbf{m}_1(\theta(\omega_M))}. \tag{5.1}$$

The q_1 lowest simulated returns are selected, where $M \geq q_1 \geq \lfloor M\alpha \rfloor$, since those estimators are most likely to contribute in the computation of the expected shortfall. The risk scenarios of these selected estimators are used to form the following set $J_1 = J_0 \cap \mathbf{m}_1([\![\theta(\omega_1),\theta(\omega_{q_1})]\!])$, where $J_0 = [\![\theta(\omega_1),\theta(\omega_M)]\!]$. The procedure from above is repeated for the risk scenarios that are in the set J_1 and so on. This is an iterative process. So suppose that the set $J_{\ell-1}$ is given, for $1 \leq \ell - 1 \leq L - 1$. The simulated returns for which the risk scenarios are in the set $J_{\ell-1}$, are already approximated with $N_{\ell-1}$ inner stage samples. Add to each of those estimators $N_\ell - N_{\ell-1}$ extra inner stage samples, such that all estimators $\left(\hat{X}_{N_\ell}^{\mathbf{m}_{\ell-1}(i)}\right)_{i\in J_{\ell-1}}$ have a total of N_ℓ inner stage samples. If $\ell \leq L-1$, we use the permutation $\mathbf{m}_\ell$ on those estimators to put them in an increasing order, $\hat{X}_{N_\ell}^{\mathbf{m}_\ell(\theta(\omega_1))} \leq \hat{X}_{N_\ell}^{\mathbf{m}_\ell(\theta(\omega_2))} \leq ... \leq \hat{X}_{N_\ell}^{\mathbf{m}_\ell(\theta(\omega_{q_{\ell-1}}))}$. Next, define the set $J_\ell = J_{\ell-1} \cap \mathbf{m}_\ell([\![\theta(\omega_1),\theta(\omega_{q_\ell})]\!])$. If $\ell = L$, we approximate the expected shortfall c_α as follows,

$$\hat{c}_\alpha = -\frac{1}{\lfloor \alpha M \rfloor} \sum_{i=1}^{\lfloor \alpha M \rfloor} \hat{X}_{N_L}^{\mathbf{m}_{L-1}(\theta(\omega_i))}. \tag{5.2}$$

5.3. Optimal allocation for q and N

In this method, in addition to the distribution of the computational cost between the number of inner stage samples N and the number of outer stage samples M, the relocation of the number of inner stage samples over the outer stage samples must also be determined. We use the same distribution as the standard Monte Carlo method defined in (4.16), to distribute the computational cost over the inner and outer stage samples. Afterwards, we determine how the number of inner stage samples will be divided over the outer stage samples. The sets $\mathbf{q} = \{q_0, q_1, ..., q_{L-1}\}$ and $\mathbf{N} = \{N_1, N_2, ..., N_L\}$ indicate how the inner stage samples will be divided over the outer stage samples. For the sets **q** and **N** must hold the following:

$$M = q_0 \geq q_1 \geq \cdots \geq q_{L-1} = \lfloor M\alpha \rfloor \quad \text{and} \quad N_1 \leq N_2 \leq \cdots \leq N_L \tag{5.3}$$

The values of the parameters in the sets **q** and **N** have a significant influence on the accuracy and the computational cost of this algorithm, so it is crucial to choose them correctly. To find the optimal choice for **q** and **N**, we must have an upper bound for the MSE of this method, for its estimate of the expected shortfall. As mentioned before, the MSE is helpful to determine whether an estimator is accurate. By setting up the MSE as a function of **q** and **N**, can we find the optimal parameters for the sets **q** and **N** by minimizing the MSE.

5.3.1. The bound on the $\mathbb{L}^p$ error

Proposition 5.3.1 provides an error bound for the approximation of the expected shortfall with the multistep algorithm that is described in section 5.2.

Proposition 5.3.1 Let $X^{\theta(\omega_1)} \leq X^{\theta(\omega_2)} \leq \cdots \leq X^{\theta(\omega_M)}$ be the returns of the portfolio, for the corresponding risk scenarios $(\theta(\omega_1), \theta(\omega_2), ..., \theta(\omega_M))$. Let $\left(\hat{X}_N^{\theta(\omega_1)}, \hat{X}_N^{\theta(\omega_2)}, ..., \hat{X}_N^{\theta(\omega_M)}\right)$ be the simulated returns of the adaptive sampling algorithm for the corresponding risk scenarios $(\theta(\omega_1), \theta(\omega_2), ..., \theta(\omega_M))$, where $N \in \{N_1, ..., N_L\}$. Then we have the following $\mathbb{L}^p$ error bound for $p \leq 1$:

$$\mathbb{E}\left[|\hat{c}_\alpha - c_\alpha|^p\right]^{\frac{1}{p}} \leq \sum_{\ell=1}^{L-1} (\delta q_\ell)^{\frac{1}{p}} \max_{(i,k)\in[\![1,\lfloor \alpha M\rfloor]\!]\times[\![q_\ell+1,M]\!]} (X^{\theta(\omega_k)} - X^{\theta(\omega_i)}) \mathbb{P}\left[\hat{X}_{N_\ell}^{\theta(\omega_k)} < \hat{X}_{N_\ell}^{\theta(\omega_i)}\right]^{\frac{1}{p}} \tag{5.4}$$

$$+ \frac{1}{\lfloor M\alpha \rfloor} \frac{\delta N_L}{N_L} \max_{1 \leq i_1 < ... < i_{\lfloor \alpha M \rfloor} \leq M} \left(\sum_{j=1}^{\lfloor \alpha M \rfloor} \mathbb{E}\left[\left|\delta \hat{X}_{N_L}^{\theta(\omega_{i_j})} - X^{\theta(\omega_{i_j})}\right|^p\right]^{\frac{1}{p}} \right) \tag{5.5}$$

$$+ \frac{1}{\lfloor M\alpha \rfloor} \frac{N_{L-1}}{N_L} \sum_{i=1}^{M} \mathbb{E}\left[\left|\hat{X}_{N_{L-1}}^{\theta(\omega_i)} - X^{\theta(\omega_i)}\right|^p\right]^{\frac{1}{p}} \tag{5.6}$$

where

$$\delta q_\ell = q_{\ell-1} - q_\ell, \quad \text{for } 1 \leq \ell \leq L \tag{5.7}$$

$$\delta N_\ell = N_{\ell-1} - N_\ell, \quad \text{for } 1 \leq \ell \leq L \tag{5.8}$$

$$\delta \hat{X}_{N_\ell}^{\theta(\omega_i)} = \frac{N_\ell \hat{X}_{N_\ell}^{\theta(\omega_i)} - N_{\ell-1} \hat{X}_{N_{\ell-1}}^{\theta(\omega_i)}}{\delta N_\ell}, \quad \text{for } 1 \leq i \leq M \tag{5.9}$$

Proof. The proof can be found in appendix A. □

The last two terms of this error bound, displayed in (5.5) and (5.6), are bounds for the error due to the standard Monte Carlo approximation. So this is the error bound for the estimate of the expected shortfall, when we would use N_L inner stage samples for each estimator $\hat{X}_{N_L}^{\theta(\omega_i)}$, where $i \in \{1, ..., M\}$. At each step of the multistep algorithm are the lowest simulated returns in the sequence of $\left(\hat{X}_{N_{\ell-1}}^i\right)_{i \in J_{\ell-1}}$ selected, where the corresponding risk scenarios are put in the set J_ℓ. The assumption is that the remaining estimators do not contribute to the computation of the expected shortfall. Consequently, the adaptive sampling algorithm assumes that the risk scenarios, who are in the set $J_{\ell-1} \backslash J_\ell$, do not contribute to the computation of the expected shortfall. The first term of the error bound, displayed in (5.4), is the bound for the error when the risk scenarios in the set $J_{\ell-1} \backslash J_\ell$ are incorrectly excluded from the computation of the expected shortfall. Thus, when there are returns in the sequence $\left(X^i\right)_{i \in J_{\ell-1} \backslash J_\ell}$, for which the value is in the lowest tail of the distribution and therefore should contribute to the computation of the expected shortfall. That is why the first term error in (5.4) contains the probability that the simulated return $\hat{X}_{N_\ell}^{\theta(\omega_k)}$ is smaller than $\hat{X}_{N_\ell}^{\theta(\omega_i)}$, while the value of $X^{\theta(\omega_i)}$ is in the lowest tail of the distribution that contributes to the expected shortfall and $X^{\theta(\omega_k)}$ is not.

The magnitude of this error bound is highly dependent on the parameters that are in the sets **q** and **N**. The error bound given in proposition 5.3.1 can be used to find the optimum choice for those sets such that the computational cost is minimized for a given error bound. The only problem is that this error bound is hard to work with. That is why we make a slight adjustment to this error bound. Before we can set up this new error bound, we first have to state the following theorem (see [5], theorem 2.1):

Theorem 5.3.2 (Bernstein's inequality) Let $X_1, ..., X_2$ be a finite sequence of independent random variables with finite variance. Denote

$$S_n = X_1 + \cdots + X_n, \qquad v_n = \frac{\mathbb{E}[X_1^2] + \cdots + \mathbb{E}[X_n^2]}{n}. \tag{5.10}$$

Assume that $\mathbb{E}[S_n] = 0$ and there exists some positive constant c such that, for any $1 \leq k \leq n$ and for any integer $d \geq 3$,

$$\mathbb{E}\left[|X_k|^d\right] \leq \frac{d!c^{d-2}}{2}\mathbb{E}\left[X_k^2\right]. \tag{5.11}$$

Then, for any positive x,

$$\mathbb{P}(S_n \geq nx) \leq \exp\left(-\frac{nx^2}{2(v_n + cx)}\right) \tag{5.12}$$

Proposition 5.3.3 states a new error bound with the help of Bernstein's inequality. This new error bound is better to work with, so it becomes easier to find the optimal parameters in the sets of **q** and **N**. This leads to the following proposition:

Proposition 5.3.3 Let $X^{\theta(\omega_1)} \leq X^{\theta(\omega_2)} \leq \cdots \leq X^{\theta(\omega_M)}$ be the returns of the portfolio, for the corresponding risk scenarios $(\theta(\omega_1), \theta(\omega_2), ..., \theta(\omega_M))$. Let $\left(\hat{X}_N^{\theta(\omega_1)}, \hat{X}_N^{\theta(\omega_2)}, ..., \hat{X}_N^{\theta(\omega_M)}\right)$ be the simulated returns of the adaptive sampling algorithm for the corresponding risk scenarios $(\theta(\omega_1), \theta(\omega_2), ..., \theta(\omega_M))$, where $N \in \{N_1, ..., N_L\}$. If there then exists an $c \in \mathbb{R}_+$, such that the random variables $Z[i,k] = \left(\hat{X}_N^{\theta(\omega_i)} - X^{\theta(\omega_i)}\right) - \left(\hat{X}_N^{\theta(\omega_k)} - X^{\theta(\omega_k)}\right)$ are satisfying Bernstein's condition, for $i, k \leq M$ and $d \geq 3$:

$$\mathbb{E}\left[|Z[i,k]|^d\right] \leq \frac{d!c^{d-2}}{2}\mathbb{E}\left[Z[i,k]^2\right] \tag{5.13}$$

Then we have that for $p \geq 1$:

$$\mathbb{E}\left[|\hat{c}_\alpha - c_\alpha|^p\right]^{\frac{1}{p}} \leq F_p(\mathbf{q}, \mathbf{N}). \tag{5.14}$$

where $F_p(\mathbf{q}, \mathbf{N})$ is equal to,

$$F_p(\mathbf{q}, \mathbf{N}) = \sum_{\ell=1}^{L-1} (\delta q_\ell)^{\frac{1}{p}} \max_{(i,k) \in [\![1, \lfloor \alpha M \rfloor]\!] \times [\![q_\ell + 1, M]\!]} (X^{\theta(\omega_k)} - X^{\theta(\omega_i)}) e^{-\frac{N_\ell \left(X^{\theta(\omega_k)} - X^{\theta(\omega_i)}\right)^2}{2p\left(\sigma_{ik}^2 + c\left(X^{\theta(\omega_k)} - X^{\theta(\omega_i)}\right)\right)}} \tag{5.15}$$

$$+ \frac{1}{\lfloor M\alpha \rfloor} \frac{\delta N_L}{N_L} \max_{1 \leq i_1 < ... < i_{\lfloor \alpha M \rfloor} \leq M} \sum_{j=1}^{\lfloor M\alpha \rfloor} \left(C_{p,\sigma} \frac{p\sigma_{i_j}^p}{(\delta N_L)^{\frac{p}{2}}} + C_{p,c} \frac{pc^p}{(\delta N_L)^p} \right)^{\frac{1}{p}} \tag{5.16}$$

$$+ \frac{1}{\lfloor M\alpha \rfloor} \frac{N_{L-1}}{N_L} \sum_{i=1}^{M} \left(C_{p,\sigma} \frac{p\sigma_i^p}{(N_{L-1})^{\frac{p}{2}}} + C_{p,c} \frac{pc^p}{(N_{L-1})^p} \right)^{\frac{1}{p}} \tag{5.17}$$

with the constant $c \in \mathbb{R}_+$ and where

$$\sigma_{ik}^2 = \mathbb{V}\left[\left(\hat{X}_{j:N}^{\theta(\omega_i)} - \hat{X}_{j:N}^{\theta(\omega_k)}\right)_{j=1}^N\right], \quad \sigma_i^2 = \mathbb{V}\left[\left(\hat{X}_{j:N}^{\theta(\omega_i)}\right)_{j=1}^N\right], \quad \text{for } i, k \leq M, \quad N \in \{N_1, ..., N_L\} \tag{5.18}$$

$$C_{p,\sigma} = 2^{p-1}\Gamma\left(\frac{p}{2}\right), \quad C_{p,c} = 4^p\Gamma(p), \quad \text{where } \Gamma \text{ is the Gamma function.} \tag{5.19}$$

Proof. The proof can be found in appendix A. □

The new upper bound error in proposition 5.3.3 has a big advantage compared to the upper bound given in proposition 5.3.1. Namely, the upper bound error in proposition 5.3.3 can be computed when the parameters of the sets $\mathbf{q}$ and $\mathbf{N}$, and the sequence of simulated returns $\left(\hat{X}_N^{\theta(\omega_1)}, \hat{X}_N^{\theta(\omega_2)}, ..., \hat{X}_N^{\theta(\omega_M)}\right)$ is given, for $N \geq 1$. This is not the case for the upper bound in proposition 5.3.1.

5.3.2. Optimization model for q and N

The allocation of $\mathbf{q}$ and $\mathbf{N}$ has a significant influence on the error bound and the computational cost of this algorithm. The trick is to determine the parameters in $\mathbf{q}$ and $\mathbf{N}$ such that there is a proper balance between the computational cost and the error bound. The total computational cost of this algorithm can be defined as follows,

$$C_{\text{adaptive}}(\mathbf{q}, \mathbf{N}) := q_0 N_1 + \sum_{\ell=1}^{L-1} q_\ell (N_{\ell+1} - N_\ell). \tag{5.20}$$

We construct an optimization problem that determines the optimal values for the parameters in $\mathbf{q}$ and $\mathbf{N}$, when it minimizes the MSE under the constraint that the computational cost is bounded by C. So at the start of this algorithm is a computational cost C given by which this method computational cost must be bounded. We have that $F_2(\mathbf{q}, \mathbf{N})$ is the MSE bound of this adaptive algorithm. The function $F_2(\mathbf{q}, \mathbf{N})$ can only be applied in an optimization model if there is a sequence of simulated returns $\left(\hat{X}_N^{\theta(\omega_1)}, \hat{X}_N^{\theta(\omega_2)}, ..., \hat{X}_N^{\theta(\omega_M)}\right)$ given. We solve this problem by generating a sequence of simulated returns $\left(\hat{X}_{N_0}^{\theta(\omega_1)}, \hat{X}_{N_0}^{\theta(\omega_2)}, ..., \hat{X}_{N_0}^{\theta(\omega_M)}\right)$, with N_0 inner stage samples. Here is $N_0 \in \mathbb{N}$ a relatively low number. This sequence is then used to determine the values of σ_{ik}^2, σ_i^2 and $X^{\theta(\omega_i)}$, for $i, k \in \{1, ..., M\}$. Since these are components in the function of $F_2(\mathbf{q}, \mathbf{N})$. Next, we obtain the following optimization problem, for finding the optimum values of $\mathbf{q}$ and $\mathbf{N}$:

$$\begin{aligned} & \underset{\mathbf{q}\ \mathbf{N}}{\text{Minimize}}\ F_2(\mathbf{q}, \mathbf{N}) \\ \text{Subject to} & \\ C_{\text{adaptive}}(\mathbf{q}, \mathbf{N}) &\leq C \\ q_\ell &\leq q_{\ell-1} \quad \text{for } \ell \in \{1, ..., L-1\} \\ N_{\ell+1} &\geq N_\ell \quad \text{for } \ell \in \{1, ..., L-1\} \\ q_\ell, N_{\ell+1} &\in \mathbb{N} \quad \text{for } \ell \in \{0, ..., L-1\} \\ q_0 &= M \\ q_{L-1} &= \lfloor M\alpha \rfloor \end{aligned} \tag{5.21}$$

The function $F_2(\mathbf{q}, \mathbf{N})$ is also dependent on a constant $c \in \mathbb{R}_+$. When solving the optimization problem in (5.21) we take $c = 0$, this is the same approach as by Bouchard, Reghai, and Virrion [6].

5.4. Numerical algorithm adaptive sampling

This algorithm can be described as an iterative process with the starting values of M, N_0, α, L and C.

Algorithm 2 Adaptive sampling method

```
procedure AdaptiveMC($M, N_0, \alpha, L, C$)
    for $i \leftarrow 1$ to $M$ do
        generate risk scenario $\theta(\omega_i)$
        for $j \leftarrow 1$ to $N_0$ do
            conditional on risk scenario $\theta(\omega_i)$, generate return $\hat{X}^{\theta(\omega_i)}_{j:N_0}$
        end for
        $\hat{X}^{\theta(\omega_i)}_{N_0} \leftarrow \frac{1}{N_0} \sum_{j=1}^{N_0} \hat{X}^{\theta(\omega_i)}_{j:N_0}$
    end for
    for $i \leftarrow 1$ to $M$ do
        $X^{\theta(\omega_i)} \leftarrow \hat{X}^{\theta(\omega_i)}_{N_0}$
        $\sigma_i^2 \leftarrow \mathbb{V}\left[\left(\hat{X}^{\theta(\omega_i)}_{j:N_0}\right)_{j=1}^{N_0}\right]$
        for $k \leftarrow 1$ to $M$ do
            $\sigma_{ik}^2 \leftarrow \mathbb{V}\left[\left(\hat{X}^{\theta(\omega_i)}_{j:N_0} - \hat{X}^{\theta(\omega_k)}_{j:N_0}\right)_{j=1}^{N_0}\right]$
        end for
    end for
    $\{\mathbf{q}, \mathbf{N}\} \leftarrow$ solve optimization problem (5.21)
    $J_0 \leftarrow \llbracket \theta(\omega_1), \theta(\omega_M) \rrbracket$
    for $\ell \leftarrow 1$ to $L$ do
        for $\theta(\omega_i) \in J_{\ell-1}$ do
            for $j \leftarrow N_{\ell-1} + 1$ to $N_\ell$ do
                conditional on risk scenario $\theta(\omega_i)$, generate return $\hat{X}^{\theta(\omega_i)}_{j:N_\ell}$
            end for
            $\hat{X}^{\theta(\omega_i)}_{N_\ell} \leftarrow \frac{1}{N_\ell} \sum_{j=1}^{N_\ell} \hat{X}^{\theta(\omega_i)}_{j:N_\ell}$
        end for
        if $\ell \le L - 1$ then
            use permutation $\mathbf{m}_\ell$ on $J_{\ell-1}$ to put the simulated returns in an increasing order,

                $$\hat{X}^{\mathbf{m}_\ell(\theta(\omega_1))}_{N_\ell} \le \hat{X}^{\mathbf{m}_\ell(\theta(\omega_2))}_{N_\ell} \le \cdots \le \hat{X}^{\mathbf{m}_\ell(\theta(\omega_{q_{\ell-1}}))}_{N_\ell}$$

            $J_\ell \leftarrow J_{\ell-1} \cap \mathbf{m}_\ell(\llbracket \theta(\omega_1), \theta(\omega_{q_\ell}) \rrbracket)$
        end if
        if $\ell = L$ then
            $\hat{c}_\alpha \leftarrow -\frac{1}{\lfloor M\alpha \rfloor} \sum_{i \in J_{\ell-1}} \hat{X}^{\mathbf{m}(\theta(\omega_i))}_{N_L}$
        end if
    end for
    return $\hat{c}_\alpha$
end procedure
```

6 Multilevel Monte Carlo method

In this section, we discuss how the Multilevel Monte Carlo (MLMC) method works and why it can be more efficient compared to the standard Monte Carlo method. In addition, we explain how the techniques occurring in the MLMC are applied in a nested simulation and how to ensure that parameters involved in the MLMC method are optimally specified. We also provide a numerical algorithm that shows how this method is applied to estimate the expected shortfall, using the correct input parameters.

6.1. Approximating the expected shortfall with the MLMC method

The MLMC method is an effective technique used to reduce the computational cost of an estimator. This method operates relatively few computations with high accuracy at a high cost and most computations with low accuracy at a low cost. Suppose that we want to approximate the expectation of the random variable c_α, so $\mathbb{E}[c_\alpha]$. Let $\hat{c}_{\alpha,\ell}$ be an estimation of c_α, that can be obtained by different amounts of computational costs and for different levels of accuracy. When the parameter ℓ in $\hat{c}_{\alpha,\ell}$ gets higher, the approximation will be more expensive to carry out, but the estimation will be more accurate. The standard Monte Carlo method is used for the approximation of $\hat{c}_{\alpha,\ell}$, with M_ℓ outer stage samples and N_ℓ inner stage samples. So the total computational cost for approximating $\hat{c}_{\alpha,\ell}$ is equal to $M_\ell \cdot N_\ell$. Let the sequence $\hat{c}_{\alpha,0},\ldots,\hat{c}_{\alpha,L-1}$, with increasing cost but also with increasing accuracy, be approximations of $\hat{c}_{\alpha,L}$. The basis of the MLMC method is based on the following identity,

$$\mathbb{E}\left[\hat{c}_{\alpha,L}\right] = \mathbb{E}\left[\hat{c}_{\alpha,0}\right] + \sum_{\ell=1}^{L} \mathbb{E}\left[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}\right]. \tag{6.1}$$

We introduce a new value G_ℓ, where G_ℓ is equal to the number of times $\hat{c}_{\alpha,\ell}$ has been approximated. So for each level ℓ, we have a sequence $\left(\hat{c}^{g}_{\alpha,\ell}\right)_{g=1}^{G_\ell}$ of estimators that approximated the expected shortfall. The approximation of $\mathbb{E}\left[\hat{c}_{\alpha,0}\right]$, is based on G_0 simulations and is given by,

$$\hat{Y}_0 = \frac{1}{G_0} \sum_{g=1}^{G_0} \hat{c}^{g}_{\alpha,0}. \tag{6.2}$$

We employ an antithetic approach for the estimation of $\mathbb{E}\left[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}\right]$. This means that the generated samples that are used for the approximation of $\hat{c}_{\alpha,\ell-1}$, will also be used for the approximation of $\hat{c}_{\alpha,\ell}$. The approximation of $\hat{c}_{\alpha,\ell-1}$ will be performed S times more often than that of $\hat{c}_{\alpha,\ell}$, where $S = M_\ell / M_{\ell-1}$. This process is explained in greater detail in section 6.3. The estimator of $\mathbb{E}\left[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}\right]$ is based on G_ℓ simulations and is given by,

$$\hat{Y}_\ell = \frac{1}{G_\ell} \sum_{g=1}^{G_\ell} \left[\hat{c}^{g}_{\alpha,\ell} - \frac{1}{S} \sum_{s=1}^{S} \hat{c}^{g,s}_{\alpha,\ell-1} \right], \quad \ell > 0. \tag{6.3}$$

The reason that we use an antithetic approach is to reduce the variance of the estimator. By applying the same generated samples for the approximation of $\hat{c}_{\alpha,\ell-1}$ and $\hat{c}_{\alpha,\ell}$, $\mathbb{V}\left[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}\right]$ will decrease. Section 6.2

explains why the decrease in variance is beneficial.

By combining (6.1),(6.2), and (6.3) we get the following estimator for the MLMC method,

$$\hat{c}_\alpha = \hat{c}_{\alpha,L} = \sum_{\ell=0}^{L} \hat{Y}_\ell = \frac{1}{G_0}\sum_{g=1}^{G_0} \hat{c}_{\alpha,0}^{g} + \sum_{\ell=1}^{L}\frac{1}{G_\ell}\sum_{g=1}^{G_\ell}\left[\hat{c}_{\alpha,\ell}^{g} - \frac{1}{S}\sum_{s=1}^{S}\hat{c}_{\alpha,\ell-1}^{g,s}\right]. \tag{6.4}$$

6.2. Complexity MLMC method

We want to ensure that the MSE is bounded by ε^2 for the approximation of the expected shortfall. As we have already seen, the MSE can be split as follows,

$$\text{MSE} = \mathbb{E}\left[(\hat{c}_\alpha - c_\alpha)^2\right] = \underbrace{\mathbb{E}\left[(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha])^2\right]}_{\text{variance}} + \underbrace{(\mathbb{E}[\hat{c}_\alpha - c_\alpha])^2}_{\text{bias}^2}, \tag{6.5}$$

where we have that

$$\mathbb{E}\left[(\hat{c}_\alpha - \mathbb{E}[\hat{c}_\alpha])^2\right] = \mathbb{V}[\hat{c}_\alpha], \qquad \mathbb{V}[\hat{c}_\alpha] = \sum_{\ell=0}^{L} G_\ell^{-1} V_\ell, \qquad V_\ell = \mathbb{V}[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}], \tag{6.6}$$

with $\hat{c}_{\alpha,-1} = 0$. An objective of the MLMC method is to distribute the computational cost such that the variance and bias squared are both bounded by $\frac{1}{2}\varepsilon^2$. This is why the application of the antithetic approach is so important. Because this decreases V_ℓ, therefore allowing G_ℓ to decrease such that the variance is still bounded by $\frac{1}{2}\varepsilon^2$. This results in a reduction of the total computational cost.

Giles also writes a general theory to show that the MLMC method is more efficient than the standard Monte Carlo method. The main theorem for the complexity of the MLMC method is stated quite generally, which means that it is applicable in many mathematical models. It does not specify which numerical approximation is used. The theorem states, conditional on certain features of the multilevel estimators and the numerical approximation, what the computational cost of the MLMC method is for a given MSE.

Theorem 6.2.1 (Complexity of the MLMC) Let c_α be a random variable, and let $\hat{c}_{\alpha,\ell}$ be the corresponding level ℓ numerical approximation. If there exist independent estimators $\hat{Y}_\ell$ based on G_ℓ Monte Carlo samples, each with expected cost C_ℓ and variance V_ℓ, and the positive constants $\kappa, \beta, \gamma, c_1, c_2, c_3$ such that $\kappa \geq \frac{1}{2}\min(\beta,\gamma)$ and

(i) $\mathbb{E}[\hat{c}_{\alpha,\ell} - c_\alpha] \leq c_1 2^{-\kappa\ell}$

(ii) $\mathbb{E}[\hat{Y}_\ell] = \begin{cases} \mathbb{E}[\hat{c}_{\alpha,0}], & \ell = 0 \\ \mathbb{E}[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}], & \ell > 0 \end{cases}$

(iii) $V_\ell \leq c_2 2^{-\beta\ell}$

(iv) $C_\ell \leq c_3 2^{\gamma\ell}$, with $C_\ell = \begin{cases} M_\ell N_\ell, & \text{if } \ell = 0 \\ M_\ell(N_\ell + N_{\ell-1}), & \text{if } \ell > 0 \end{cases}$

then there exists a positive constant c_4 such that for any $\varepsilon < e^{-1}$, there are values L and G_ℓ for which the multilevel estimator

$$\hat{c}_\alpha = \sum_{\ell=0}^{L} \hat{Y}_\ell \tag{6.7}$$

has a mean square error with bound

$$\text{MSE} = \mathbb{E}\left[(\hat{c}_\alpha - c_\alpha)^2\right] < \varepsilon^2 \tag{6.8}$$

with a computational complexity C with bound

$$C \leq \begin{cases} c_4\varepsilon^{-2}, & \beta > \gamma, \\ c_4\varepsilon^{-2}(\log(\varepsilon))^2, & \beta = \gamma, \\ c_4\varepsilon^{-2-(\gamma-\beta)/\kappa}, & \beta < \gamma. \end{cases} \tag{6.9}$$

See [12] for a complete proof.

Recall that it is already examined in the literature that the total computational cost for the standard Monte Carlo method in a nested simulation is equal to $\mathcal{O}(\varepsilon^{-3})$, to get an upper bound of ε^2 for the MSE. Theorem 6.2.1 shows that the MLMC method can be more efficient for the correct parameters κ, β and γ. Those parameters, which are given in the conditions (i), (iii), and (iv) of Theorem 6.2.1 depend on the values M_ℓ and N_ℓ. Theorem 6.2.1 states that the computational cost of the MLMC method depends on the parameters κ, β, and γ. The inconvenience is that the values κ and β are difficult to derive analytically. For this reason, we try different strategies for the values N_ℓ and M_ℓ to determine what the best approach is. The values M_ℓ and N_ℓ always grow along with powers of 2. We denote the strategy in the following manner,

$$\begin{aligned}
"M_1": &\quad M_\ell = M_0 2^\ell &\quad N_\ell &\equiv N_0\\
"M_1N_1": &\quad M_\ell = M_0 2^\ell &\quad N_\ell &= N_0 2^\ell\\
"M_2N_1": &\quad M_\ell = M_0 2^{2\ell} &\quad N_\ell &= N_0 2^\ell\\
"M_2N_2": &\quad M_\ell = M_0 2^{2\ell} &\quad N_\ell &= N_0 2^{2\ell}\\
&\quad\vdots
\end{aligned}$$

6.3. The antithetic approach

The antithetic approach as described by Giles in [13], is a technique that is used in Monte Carlo methods to reduce the variance. In the MLMC method is the antithetic approach used for approximating $\mathbb{E}[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}]$. We start with the approximation of $\hat{c}_{\alpha,\ell-1}$. First, we generate the sequence $(\theta(\omega_m))_{m=1}^{M_{\ell-1}}$ of i.i.d. risk scenarios. Conditional on each risk scenario $\theta(\omega_i)$, with $i \in \{1,...,M_{\ell-1}\}$, we generate a number of $N_{\ell-1}$ i.i.d. samples $\hat{X}_{j:N_{\ell-1}}^{\theta(\omega_i)}$, where $j \in \{1,...,N_{\ell-1}\}$. Those samples are used to compute $\hat{X}_{N_{\ell-1}}^{\theta(\omega_i)}$, for all $i \in \{1,...,M_{\ell-1}\}$,

$$\hat{X}_{N_{\ell-1}}^{\theta(\omega_i)} = \frac{1}{N_{\ell-1}} \sum_{j=1}^{N_{\ell-1}} \hat{X}_{j:N_{\ell-1}}^{\theta(\omega_i)}. \tag{6.10}$$

This gives us the sequence $\left(\hat{X}_{N_{\ell-1}}^{\theta(\omega_1)}, \hat{X}_{N_{\ell-1}}^{\theta(\omega_2)}, ..., \hat{X}_{N_{\ell-1}}^{\theta(\omega_{M_{\ell-1}})}\right)$ of simulated returns. Those returns are used to make an approximation of the expected shortfall, as explained in section 4. The procedure form above is repeated S times, where for S holds that $S = M_\ell / M_{\ell-1}$. The estimate $\hat{c}_{\alpha,\ell-1}$ is based on the average expected shortfall of those S approximations.

Remember, that the generated samples which are used for the approximation of $\hat{c}_{\alpha,\ell-1}$, will also be used for the approximation of $\hat{c}_{\alpha,\ell}$. To approximate $\hat{c}_{\alpha,\ell}$, we put the already generated $S \times M_{\ell-1}$ risk scenarios together such that we have a total of M_ℓ risk scenarios. The corresponding simulated returns of those risk scenarios are already approximated with $N_{\ell-1}$ inner stage samples. We add to those simulated returns $N_\ell - N_{\ell-1}$ more inner stage samples such that all returns are approximated with N_ℓ inner stage samples. This gives us the sequence $\left(\hat{X}_{N_\ell}^{\theta(\omega_1)}, \hat{X}_{N_\ell}^{\theta(\omega_2)}, ..., \hat{X}_{N_\ell}^{\theta(\omega_{M_\ell})}\right)$. We use this sequence to compute $\hat{c}_{\alpha,\ell}$, as explained in section 4. Algorithm 3 shows how the procedure from above is performed with the given parameters $M_\ell, M_{\ell-1}, N_\ell, N_{\ell-1}$ and α.

Algorithm 3 Antithetic approach

```
procedure Antithetic($M_\ell, N_\ell, M_{\ell-1}, N_{\ell-1}, \alpha$)
    $S \leftarrow M_\ell / M_{\ell-1}$
    for $s \leftarrow 1$ to $S$ do
        for $i \leftarrow 1$ to $M_{\ell-1}$ do
            generate risk scenario $\theta^s(\omega_i)$
            for $j \leftarrow 1$ to $N_{\ell-1}$ do
                conditional on risk scenario $\theta^s(\omega_i)$, generate return $\hat{X}^{\theta^s(\omega_i)}_{j:N_{\ell-1}}$
            end for
            $\hat{X}^{\theta^s(\omega_i)}_{N_{\ell-1}} \leftarrow \frac{1}{N_{\ell-1}} \sum_{j=1}^{N_{\ell-1}} \hat{X}^{\theta^s(\omega_i)}_{j:N_{\ell-1}}$
            for $j \leftarrow N_{\ell-1}+1$ to $N_\ell$ do
                conditional on risk scenario $\theta^s(\omega_i)$, generate return $\hat{X}^{\theta^s(\omega_i)}_{j:N_\ell}$
            end for
            $\hat{X}^{\theta^s(\omega_i)}_{N_\ell} \leftarrow \frac{1}{N_\ell} \sum_{j=1}^{N_\ell} \hat{X}^{\theta^s(\omega_i)}_{j:N_\ell}$
        end for
        use permutation $\mathbf{m}_s$ on $[\![\theta^s(\omega_1), \theta^s(\omega_{M_{\ell-1}})]\!]$ to put the simulated returns in an increasing order,

            $\hat{X}^{\mathbf{m}_s(\theta^s(\omega_1))}_{N_{\ell-1}} \leq \hat{X}^{\mathbf{m}_s(\theta^s(\omega_2))}_{N_{\ell-1}} \leq \cdots \leq \hat{X}^{\mathbf{m}_s(\theta^s(\omega_{M_{\ell-1}}))}_{N_{\ell-1}}$

        $\hat{c}^s_{\alpha,\ell-1} \leftarrow -\frac{1}{\lfloor M_{\ell-1}\alpha \rfloor} \sum_{i=1}^{\lfloor M_{\ell-1}\alpha \rfloor} \hat{X}^{\mathbf{m}_s(\theta^s(\omega_i))}_{N_{\ell-1}}$
    end for
    use permutation $\mathbf{m}$ on $[\![\theta(\omega_1), \theta(\omega_{M_\ell})]\!]$ to put the simulated returns $\hat{X}^{\theta^s(\omega_i)}_{N_\ell}$, for all $s \in \{1, ..., S\}$ and
for all $i \in \{1, ..., M_{\ell-1}\}$, in an increasing order,

            $\hat{X}^{\mathbf{m}(\theta(\omega_1))}_{N_\ell} \leq \hat{X}^{\mathbf{m}(\theta(\omega_2))}_{N_\ell} \leq \cdots \leq \hat{X}^{\mathbf{m}(\theta(\omega_{M_\ell}))}_{N_\ell}$

    $\hat{c}_{\alpha,\ell} \leftarrow -\frac{1}{\lfloor M_\ell \alpha \rfloor} \sum_{i=1}^{\lfloor M_\ell \alpha \rfloor} \hat{X}^{\mathbf{m}(\theta(\omega_i))}_{N_\ell}$
    $\hat{c}_{\alpha,\ell-1} \leftarrow \frac{1}{S} \sum_{s=1}^{S} \hat{c}^s_{\alpha,\ell-1}$
    return $\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}$
end procedure
```

As already explained is the expectation $\mathbb{E}[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}]$ based on G_ℓ simulations. So to compute $\mathbb{E}[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}]$ is algorithm 3 performed G_ℓ times. The total computational cost for obtaining this estimator is equal to $G_\ell(M_\ell N_\ell + S M_{\ell-1} N_{\ell-1}) = G_\ell M_\ell (N_\ell + N_{\ell-1})$. Therefore, the total computational cost for the MLMC method is equal to,

$$C_{\text{MLMC}} = \sum_{\ell=0}^{L} G_\ell M_\ell (N_\ell + N_{\ell-1}) \tag{6.11}$$

where $N_{-1} = 0$.

6.4. Optimal G_ℓ

The value of G_ℓ indicates how many times $\hat{c}_{\alpha,\ell}$ is approximated. For the MLMC method holds that when the computational cost of the estimator gets higher, the less often its approximation is performed. Since the computational cost of $\hat{c}_{\alpha,\ell}$ only increases as ℓ increases, will G_ℓ only decrease when ℓ increases. The values $(\mathbb{E}[\hat{c}_\alpha - c_\alpha])^2$ and $\mathbb{V}[\hat{c}_\alpha]$ must both be bounded by $\frac{1}{2}\varepsilon^2$. The bias squared does not depend on how we choose G_ℓ, for each level $\ell \in \{0, ..., L\}$. So we can determine each level for G_ℓ without interfering with the inequality $(\mathbb{E}[\hat{c}_\alpha - c_\alpha])^2 \leq \frac{1}{2}\varepsilon^2$. When choosing G_ℓ, where $\ell \in \{0, ..., L\}$, we only have to take into account that $\mathbb{V}[\hat{c}_\alpha]$ is bounded by $\frac{1}{2}\varepsilon^2$. All the levels of G_ℓ are picked such that the computational cost is minimized. We use the Lagrange multiplier method for finding the optimal value for G_ℓ, for all $\ell \in \{0, ..., L\}$. We want to minimize the computational cost C of the MLMC method. We can write C as a function of G_ℓ,

$$C = \sum_{l=0}^{L} G_\ell C_\ell, \quad \text{with } C_\ell = \begin{cases} M_\ell N_\ell, & \text{if } \ell = 0 \\ M_\ell (N_\ell + N_{\ell-1}), & \text{if } \ell > 0 \end{cases}. \tag{6.12}$$

The constraint function in the Lagrange multiplier method must also contain all the levels of G_ℓ, so we use the following Lagrangian function,

$$\mathcal{L}(G_0, ..., G_L, \lambda) = \sum_{l=0}^{L} C_\ell G_\ell - \lambda \left(V - \sum_{\ell=0}^{L} G_\ell^{-1} V_\ell \right), \tag{6.13}$$

with $V = \mathbb{V}[\hat{c}_\alpha]$. Now we are going to find the stationary points of $\mathcal{L}(G_0, ..., G_L, \lambda)$, where we treat G_ℓ for all $\ell \in \{0, ..., L\}$ as a continuous variable:

$$\frac{\delta \mathcal{L}}{\delta G_\ell} = 0 \iff C_\ell - \lambda V_\ell G_\ell^{-2} = 0 \iff G_\ell = \sqrt{\lambda V_\ell C_\ell^{-1}} \qquad \forall \ell \in \{0, ..., L\} \tag{6.14}$$

$$\frac{\delta \mathcal{L}}{\delta \lambda} = 0 \iff V - \sum_{\ell=0}^{L} G_\ell^{-1} V_\ell = 0 \tag{6.15}$$

By combining them we can isolate λ,

$$V - \sum_{\ell=0}^{L} G_\ell^{-1} V_\ell = 0 \quad \iff \quad V - \sum_{\ell=0}^{L} V_\ell \sqrt{\lambda^{-1} V_\ell^{-1} C_\ell} = 0 \tag{6.16}$$

$$\iff \quad \lambda^{-1/2} \sum_{\ell=0}^{L} \sqrt{V_\ell C_\ell} = V \tag{6.17}$$

$$\iff \quad \lambda = \left(\frac{\sum_{\ell=0}^{L} \sqrt{V_\ell C_\ell}}{V} \right)^2 \tag{6.18}$$

Finally we can find the optimal value for all G_ℓ, where $\ell \in \{0, ..., L\}$,

$$G_\ell = \sqrt{\lambda V_\ell C_\ell^{-1}} \quad \iff \quad G_\ell = \frac{1}{V} \sqrt{V_\ell C_\ell^{-1}} \sum_{\ell=0}^{L} \sqrt{V_\ell C_\ell} \tag{6.19}$$

Substituting the upper bound $\frac{1}{2}\varepsilon^2$ for the variance V, gives us the following optimal integer for all levels of G_ℓ, where $\ell \in \{0, ..., L\}$:

$$G_\ell = \left\lceil 2\varepsilon^{-2} \sqrt{V_\ell C_\ell^{-1}} \sum_{\ell=0}^{L} \sqrt{V_\ell C_\ell} \right\rceil, \tag{6.20}$$

where $\lceil x \rceil$ denotes the smallest integer larger than or equal to x.

6.5. Iterative process of the MLMC

As shown in equation (6.4), the estimated expected shortfall is dependent on the value L. The value L is not known before the beginning of the MLMC method. The MLMC method has an iterative process that determines the correct value for L. The value L is picked such that the bias squared is bounded by $\frac{1}{2}\varepsilon^2$. During the process, the MLMC method checks if the bias squared of the estimator is bounded by $\frac{1}{2}\varepsilon^2$. The process starts over when this is not the case and increases the value L by 1. When the bias squared of the estimator is bounded by $\frac{1}{2}\varepsilon^2$, stops the iterative process.

6.5.1. Bias estimation

The optimal value for L can be obtained by iteratively testing if the bias squared has an upper bound of $\frac{1}{2}\varepsilon^2$. The expectation $\mathbb{E}[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}]$, at each level $\ell \in \{1, ..., L\}$, can give information about the bias of the estimator. Giles [13] proposes to use the following convergence condition,

$$\mathbb{E}[\hat{c}_{\alpha,\ell} - c_\alpha] \approx c M_\ell^{-1}, \tag{6.21}$$

for some constant c. From (6.21) follows that

$$\begin{aligned}\mathbb{E}\left[\hat{c}_{\alpha,\ell}-\hat{c}_{\alpha,\ell-1}\right] =& \mathbb{E}\left[\hat{c}_{\alpha,\ell}-c_{\alpha}\right]-\mathbb{E}\left[\hat{c}_{\alpha,\ell-1}-c_{\alpha}\right]\\ \approx& c\left(M_{\ell}^{-1}-M_{\ell-1}^{-1}\right)\\ =& cM_{\ell}^{-1}(1-S)\\ \approx& \mathbb{E}\left[\hat{c}_{\alpha,\ell}-c_{\alpha}\right](1-S) \end{aligned} \tag{6.22}$$

Thus, the bias of $\hat{c}_{\alpha}$ can be approximated by

$$\mathbb{E}\left[\hat{c}_{\alpha}-c_{\alpha}\right]=\mathbb{E}\left[\hat{c}_{\alpha,L}-c_{\alpha}\right]\approx\mathbb{E}\left[\hat{c}_{\alpha,L}-\hat{c}_{\alpha,L-1}\right](1-S)^{-1} \tag{6.23}$$

This can be used for approximating the upper bound of the bias. The bias squared $\left(\mathbb{E}[\hat{c}_{\alpha}-c_{\alpha}]\right)^2$ must be bounded by $\frac{1}{2}\varepsilon^2$, so we have that the inequality $\mathbb{E}[\hat{c}_{\alpha}-c_{\alpha}]\leq\varepsilon/\sqrt{2}$ must hold. The iterative process then determines the optimal value for L, by increasing the value L until the following condition holds,

$$\left|\hat{Y}_L\right|\leq\frac{1}{\sqrt{2}}(S-1)\varepsilon. \tag{6.24}$$

Since equation (6.23) does not always hold, are we being more cautious and use the following equation when we want to test the convergence of the bias,

$$\max\left\{S^{-1}\left|\hat{Y}_{L-1}\right|,\left|\hat{Y}_L\right|\right\}<\frac{1}{\sqrt{2}}(S-1)\varepsilon. \tag{6.25}$$

The optimal value of G_ℓ, for all $\ell\in\{0,\dots,L\}$, is dependent on the value L. So we update the value G_ℓ during the iterative process. Since G_ℓ is also used to compute Y_ℓ by which the convergence of the bias is tested.

6.6. Numerical algorithm MLMC method

The Numerical algorithm of the MLMC method makes use of algorithms 1 and 3, which are noted as StandardMC(·) and Antithetic(·). The expected shortfall is approximated with the MLMC method as follows, for the starting values of M_0, N_0, G_0, α and ε:

Algorithm 4 Multilevel Monte Carlo method

```
procedure MLMC($M_0, N_0, G_0, \alpha, \varepsilon$)
    $L \leftarrow 0$
    $S \leftarrow M_1/M_0$
    $G \leftarrow G_0$
    $Y \leftarrow$ not converged
    while $Y \leftarrow$ not converged do
        for $g \leftarrow 1$ to $G$ do
            $\hat{c}^{g}_{\alpha,L} \leftarrow \begin{cases} \text{StandardMC}(M_0, N_0, \alpha), & \textbf{if } L = 0 \\ \text{Antithetic}(M_L, N_L, M_{L-1}, N_{L-1}, \alpha), & \textbf{if } L > 0 \end{cases}$
        end for
        $V_L \leftarrow \mathbb{V}\left[\left(\hat{c}^{g}_{\alpha,\ell}\right)_{g=1}^{G}\right]$
        $C_L \leftarrow \begin{cases} M_0 N_0, & \textbf{if } L = 0 \\ M_L(N_L + N_{L-1}), & \textbf{if } L > 0 \end{cases}$
        $G_L^{\text{old}} \leftarrow 0$
        for $\ell \leftarrow 0$ to $L$ do
            $G_\ell^{\text{new}} \leftarrow \left\lceil 2\varepsilon^{-2}\sqrt{V_\ell C_\ell^{-1}} \sum_{\ell=0}^{L} \sqrt{V_\ell C_\ell} \right\rceil$
            for $g \leftarrow G_\ell^{\text{old}} + 1$ to $G_\ell^{\text{new}}$ do
                $\hat{c}^{g}_{\alpha,\ell} \leftarrow \begin{cases} \text{StandardMC}(M_0, N_0, \alpha), & \textbf{if } \ell = 0 \\ \text{Antithetic}(M_\ell, N_\ell, M_{\ell-1}, N_{\ell-1}, \alpha), & \textbf{if } \ell > 0 \end{cases}$
            end for
            $\hat{Y}_\ell \leftarrow \frac{1}{G_\ell^{\text{new}}} \sum_{g=1}^{G_\ell^{\text{new}}} \hat{c}^{g}_{\alpha,\ell}$
            $V_\ell \leftarrow \mathbb{V}\left[\left(\hat{c}^{g}_{\alpha,\ell}\right)_{g=1}^{G_\ell^{\text{new}}}\right]$
            $C_\ell \leftarrow \begin{cases} M_\ell N_\ell, & \textbf{if } \ell = 0 \\ M_\ell(N_\ell + N_{\ell-1}), & \textbf{if } \ell > 0 \end{cases}$
            $G_\ell^{\text{old}} \leftarrow G_\ell^{\text{new}}$
        end for
        $G \leftarrow G_L^{\text{new}}$
        if $L \geq 2$ and $\max\left\{S^{-1}\left|\hat{Y}_{L-1}\right|, \left|\hat{Y}_L\right|\right\} < \frac{1}{\sqrt{2}}(S-1)\varepsilon$ then
            $Y \leftarrow$ converged
            $\hat{c}_\alpha \leftarrow \sum_{\ell=0}^{L} \frac{1}{G_\ell^{\text{new}}} \sum_{g=1}^{G_\ell^{\text{new}}} \hat{c}^{g}_{\alpha,\ell}$
        end if
        $L \leftarrow L + 1$
    end while
    return $\hat{c}_\alpha$
end procedure
```

7 Verification

In this section, we check with the help of numerical experiments if the theories that were discussed in sections 4, 5 and 6 also hold in practice. First, we start by defining our starting parameters such that we can generate the inner and outer stages, as described in section 3.3. Next, we verify whether the theories are correct for the standard Monte Carlo, adaptive sampling and the MLMC method, which are discussed in sections 4, 5 and 6.

7.1. Simulation setup

In section 3, we have explained how the inner and outer stage samples are generated in the simulation. In this section, we want to perform numerical tests to observe and analyze how each method performs. Before we can do that, we have to determine the parameters of the inner and outer stage samples. Table 7.1 gives an overview of these parameters.

S_0	100
μ	0.04
σ	0.2
r	0.07
K	90
τ	0.1
T	0.25

Table 7.1: Starting parameters simulation.

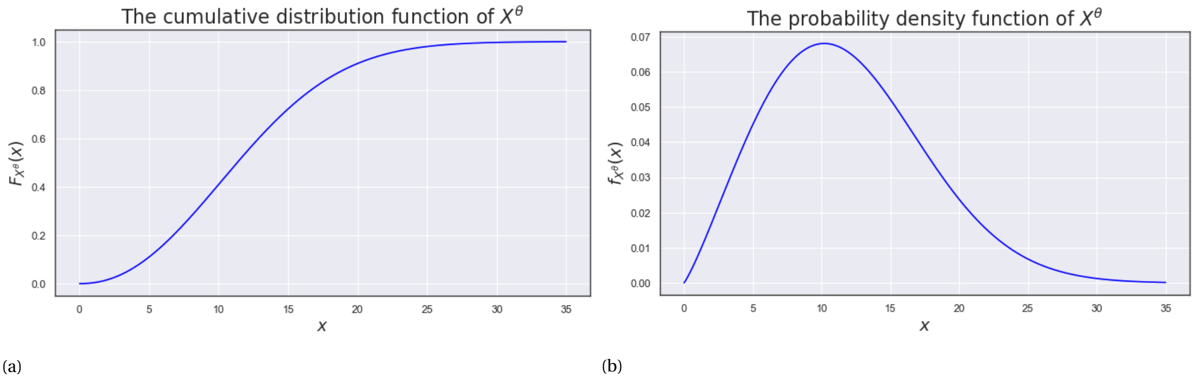
We can use these parameters to display the distribution of X^θ. In section 3.3.1, we had derived that the CDF of X^θ is equal to,

$$F_{X^\theta}(x) = \mathbb{P}\left(X^\theta \leq x\right) = F_{\mathcal{N}(0,1)}\left(\frac{\log \frac{V_c^{-1}(x)}{S_0} - \left(\mu - \sigma^2/2\right)\tau}{\sigma\sqrt{\tau}}\right). \tag{7.1}$$

The inverse of the Black-Scholes formula can be computed by applying the Newton-Raphson method, which is a root-finding algorithm. The probability density function (PDF) gives an impression of how frequent a value of the portfolio return X^θ occurs. The PDF of X^θ can be computed as follows,

$$f_{X^\theta}(x) = \frac{\delta F_{X^\theta}(x)}{\delta x} \approx \frac{F_{X^\theta}(x+h) - F_{X^\theta}(x)}{h}, \tag{7.2}$$

with a small value for $h > 0$. Figure 3 displays the CDF and PDF of X^θ.



(a) (b)

Figure 7.1: Illustration of the distribution of X^θ, where Figure 7.1a displays the CDF of X^θ and Figure 7.1b displays the PDF of X^θ.

The confidence levels of the expected shortfall that we have selected to approximate are 0.05, 0.1, and 0.2, so $\alpha \in \{0.05, 0.1, 0.2\}$. The corresponding values of the expected shortfall can be computed by equation (3.18), and are equal to -4.5284, -3.2425 and -2.3388.

7.2. Standard Monte Carlo method

For the standard Monte Carlo method, we must select the right ratio between the number of outer stage samples M and inner stage samples N, for a given computational cost C. The optimal ratio given in section 4, is that $M = \left(\frac{\gamma_1}{2\gamma_2}\right)^{1/3} C^{2/3}$ and $N = \left(\frac{2\gamma_2}{\gamma_1}\right)^{1/3} C^{1/3}$. When a nested simulation has to be applied to estimate the expected shortfall, it is unclear how γ_1 and γ_2 should be computed. Because if this is the case, the expected shortfall can also be determined analytically. Therefore, we do not compute the values of γ_1 and γ_2 before applying the nested simulation. Fortunately, γ_1 and γ_2 are constants and their values become insignificant when C gets large enough. Figure 7.2 illustrates this clearly. It shows what the MSE of the standard Monte Carlo method is for different values of p, where the number of outer stage samples is equal to $M = C^p$, and the number of inner stage samples is equal to $N = C^{1-p}$. Here, is the computational cost C equal to 10^7.

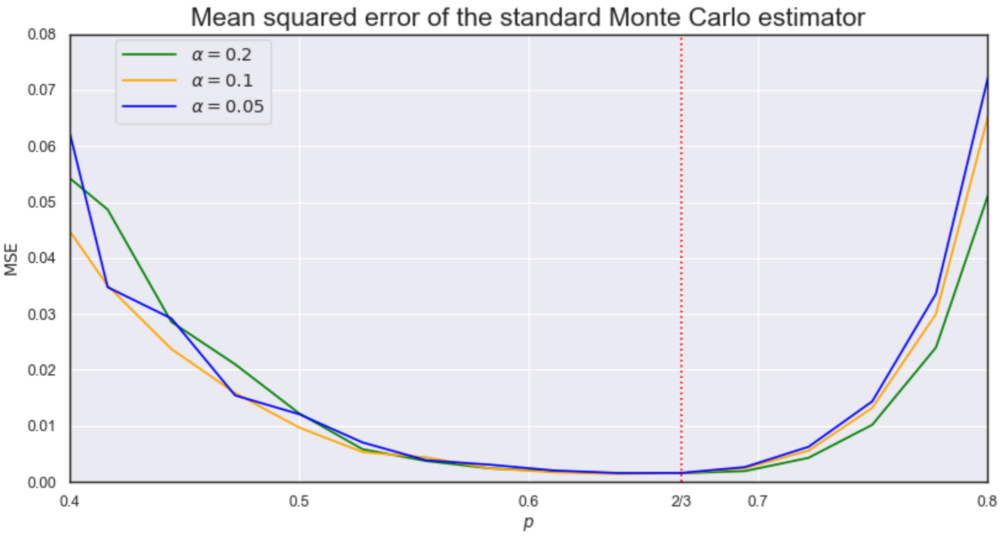


Figure 7.2: The MSE of the standard Monte Carlo method, with $M = C^p$, $N = C^{1-p}$, and $C = 10^7$.

Figure 7.2 displays that the MSE is highly influenced by the spread of the number of outer and inner stage samples that are used for the approximation of the expected shortfall. In addition, Figure 7.2 shows that the MSE is minimal at $p = 2/3$. This is equivalent to the optimum value that is given in section 4 with $\frac{2\gamma_2}{\gamma_1} = 1$. Therefore, to apply the standard Monte Carlo method optimally in this simulation, γ_1 and γ_2 do not need to be computed in advance and for a given computational cost C, the number of outer stage samples M is equal to $C^{2/3}$, and the number of inner stage samples N is equal to $C^{1/3}$.

7.3. Adaptive sampling method

The idea behind the adaptive sampling method is to relocate the inner stage samples such that the simulated returns who approximate the expected shortfall contain the most inner stage samples. It is therefore important that the optimization problem stated in (5.21) works correctly, because it determines how the inner stage samples are distributed over the outer stage samples. This optimization problem's result depends on the error bound for the approximation of the expected shortfall, stated in proposition 5.3.3. Since the optimization problem minimizes the error bound stated in proposition 5.3.3. By examining the allocation of the optimization problem in (5.21), we get an indication if this error can be used to obtain a reasonable allocation of the inner stage samples.

The optimization problem in (5.21) can become very complex for a large value of L and therefore taking a long time to solve. We have chosen in this thesis to set $L = 3$. We solved the optimization problem in (5.21) for $C = 10^7$ and with $M = C^{2/3}$, for $\alpha \in \{0.05, 0.1, 0.2\}$. Figure 7.3 shows how this optimization problem distributes the inner stage samples over the simulated returns $\hat{X}^{\theta(\omega_i)}$, where $i \in \{1,...,M\}$. The black dotted line in Figure 7.3, illustrates the amount of inner stage samples the standard Monte Carlo method uses for each simulated return, which is equal to $C^{1/3}$. For the simulated returns holds that $\hat{X}^{\theta(\omega_1)} \leq \hat{X}^{\theta(\omega_2)} \leq \cdots \leq \hat{X}^{\theta(\omega_M)}$.

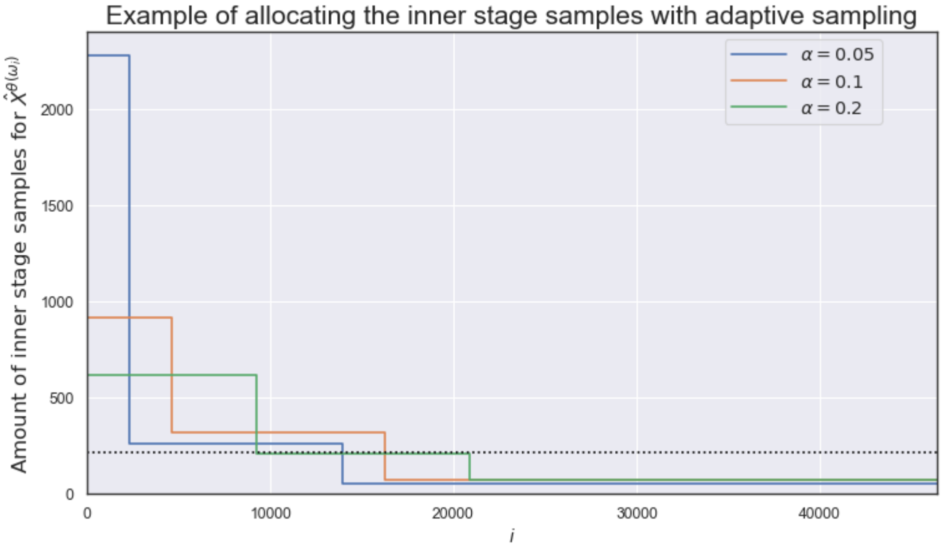


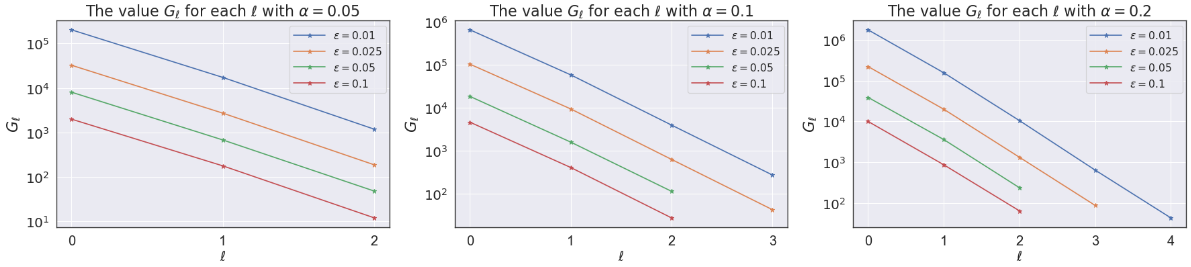
Figure 7.3: Illustration of the allocation of inner stage samples for $\hat{X}^{\theta(\omega_i)}$ with adaptive sampling compared to the standard Monte Carlo method, where for the simulated returns holds that $\hat{X}^{\theta(\omega_1)} \leq \hat{X}^{\theta(\omega_2)} \leq \cdots \leq \hat{X}^{\theta(\omega_M)}$.

Intuitively, it appears correct how the optimization problem in (5.21) distributes the inner stage samples. The lower the value of the simulated return $\hat{X}^{\theta(\omega_i)}$, the more inner stage samples $\hat{X}^{\theta(\omega_i)}$ contains, for $i \in \{1,..,M\}$. We can observe that the adaptive sampling method uses a lot more inner stage samples, for the computation of the simulated returns that are used to estimate the expected shortfall, in comparison with the standard Monte Carlo method. This must also be the case, since the adaptive sampling method aims to allocate the inner stage samples such that the simulated returns that are used to estimate the expected shortfall, are the most accurate.

7.4. Multilevel Monte Carlo method

As discussed in section 6.2, multiple strategies can be applied in the MLMC method, where a strategy implies the amount by which the values of N_ℓ and M_ℓ increase when the value ℓ increases. After repeatedly testing different strategies, we concluded that the strategy '$M2N2$' provides the best results for the simulation. This strategy implies that $N_\ell = N_0 \cdot 2^{2\ell}$ and $M_\ell = M_0 \cdot 2^{2\ell}$. Since we are considering the lowest 20%, 10% and 5% of the scenarios, M_0 must be at least 5, 10 or 20. Otherwise, $\mathbb{E}[\hat{c}_{\alpha,0}]$ cannot be computed. Therefore, the starting parameters for $\alpha = 0.2$ are $M_0 = 5$ and $N_0 = 3$, for $\alpha = 0.1$ are the starting parameters $M_0 = 10$ and $N_0 = 5$ and for $\alpha = 0.05$ are the starting parameters $M_0 = 20$ and $N_0 = 10$.

To demonstrate how the MLMC method performs, we approximate the expected shortfall with multiple values for ε. The different values for G_ℓ are shown in Figure 7.4. Recall that G_ℓ expresses how often $\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}$ is estimated, where $\hat{c}_{\alpha,-1} = 0$. The fact that G_ℓ decreases as ℓ increases is therefore obvious, as the MLMC method decreases the number of approximations when an estimate becomes more expensive to perform. Figure 7.4 also shows that when the value of ε decreases, more approximations are carried out. This is due the fact on how G_ℓ is defined in (6.20). Figure 7.4 also shows that the value L is not constant for each estimator. In section 6.5 we explained that the value L is increased by 1 if the bias squared of the estimator is not yet bounded by $\frac{1}{2}\varepsilon^2$. Therefore, for a lower value of ε, the value for L will increase since the upper bound of $\frac{1}{2}\varepsilon^2$ is much lower. Figure 7.4 also illustrates that α affects the value of L, since the value for L is higher when α is higher. This is because for a higher value of α, M_ℓ and N_ℓ are lower and the computation of $\mathbb{E}[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}]$ is less accurate. As a result, the estimator's bias squared is not yet bounded by $\frac{1}{2}\varepsilon^2$ for the same value of L.



(a)

(b)

(c)

Figure 7.4: The amount of times $\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}$ is approximated for different values of ℓ, in Figure 7.4a for $\alpha = 0.05$, in Figure 7.4b for $\alpha = 0.1$ and in Figure 7.4c for $\alpha = 0.2$.

Figure 7.5 displays the computational cost for different values of ε. We can observe that $\mathcal{O}(\varepsilon^{-3})$ is an upper bound for the computational cost of the MLMC method. In the literature was discussed that $\mathcal{O}(\varepsilon^{-3})$ the best obtainable upper bound was for the standard Monte Carlo method in a nested simulation. Therefore, the MLMC method appears to be more efficient than the standard Monte Carlo method. Figure 7.5 also shows that the differences between the computational cost ratios become significantly smaller when the value of ε decreases. So for a small value of epsilon, the initial values of M_0 and N_0 become less relevant.

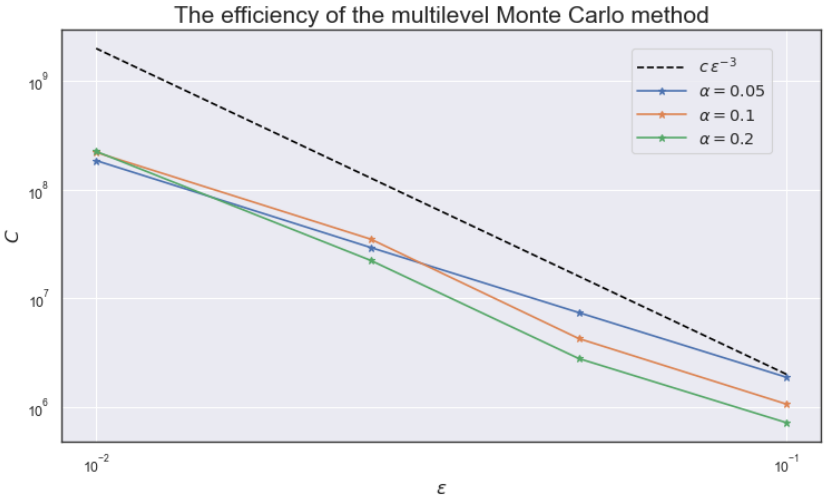
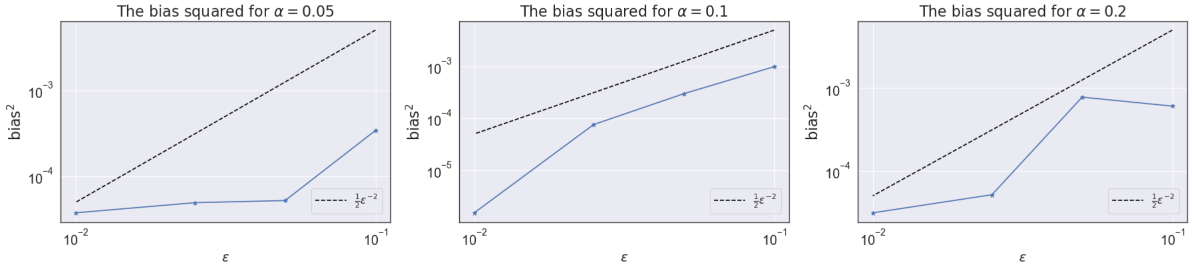


Figure 7.5: Testing the efficiency of the MLMC method by plotting the computational cost versus the accuracy of the estimators, for $\alpha \in \{0.05, 0.1, 0.2\}$.

As explained in section 6.2, the MLMC method operates such that both the bias squared and the variance are bounded by $\frac{1}{2}\varepsilon^2$. This is necessary because the MSE is equal to the sum of these two components, which determines the estimator's accuracy. To check whether this is true, we applied the MLMC method several times to estimate the expected shortfall, for $\alpha \in \{0.05, 0.1, 0.2\}$ and $\varepsilon \in \{0.01, 0.025, 0.05, 0.1\}$. The results were used to determine the bias squared and the variance from the estimates of the MLMC method.

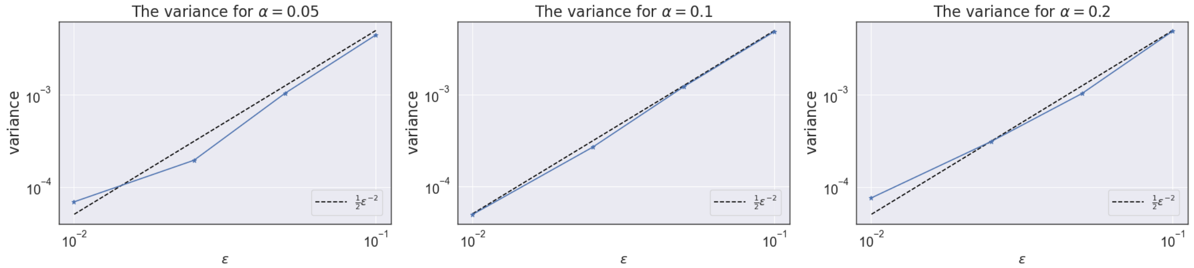
Figure 7.6 shows that for each value of α and each value of ε, the bias squared is bounded by $\frac{1}{2}\varepsilon^2$. What is also interesting to note is that the bias squared does not decrease smoothly. This is because the bias squared of the estimator also depends on the value L, and the value L is not continuous. As illustrated in Figure 7.4, it holds that $L \in \{2, 3, 4\}$. Therefore, the decrease of the bias squared for a lower value of ε shall be less regular.



(a) (b) (c)

Figure 7.6: Numerical results of the bias squared obtained by multiple estimates of the MLMC method approximating the expected shortfall, in Figure 7.6a for $\alpha = 0.05$, in Figure 7.6b for $\alpha = 0.1$ and in Figure 7.6b for $\alpha = 0.2$.

Figure 7.7 shows the variance of the MLMC estimator for each value of α and ε. What is interesting is that variance decreases smoothly alongside the value of ε. This is related to the way G_ℓ is defined in (6.20). Since this value of G_ℓ ensures that the value $\mathbb{E}[\hat{c}_{\alpha,\ell} - \hat{c}_{\alpha,\ell-1}]$ is computed exactly enough times, for $\ell \in \{0, ..., L\}$, so that the variance of the estimator is just bounded by $\frac{1}{2}\varepsilon^2$. Another notable result is that for $\varepsilon = 0.05$, for both $\alpha = 0.2$ and $\alpha = 0.05$ is the estimator's variance above the boundary of $\frac{1}{2}\varepsilon^2$. This can be explained by the fact that multiple values of L occur in the estimators when this approximation is carried out. This results in a larger variation between the values of the estimators. If the estimators with the same value for L are added together, then the variance of those estimators are bounded by $\frac{1}{2}\varepsilon^2$.



(a) (b) (c)

Figure 7.7: Numerical results of variance obtained by multiple estimates of the MLMC method approximating the expected shortfall, in Figure 7.7a for $\alpha = 0.05$, in Figure 7.7b for $\alpha = 0.1$ and in Figure 7.7b for $\alpha = 0.2$.

8
Numerical results

In this section, we provide an overall comparison of the efficiency of the standard Monte Carlo, adaptive sampling and the MLMC method. We estimate the expected shortfall several times with each method. The results of each method are used to determine the bias squared, variance and the MSE of the estimators. First, we explain what the input parameters are for each method, in our numerical experiment. Afterwards, we present the results of the simulation.

8.1. Input parameters

We want to select the input parameters in a way that each method is applied optimally, such that they can be compared fairly. For each method, we use the same amount of computational cost. The only problem is that the MLMC method does not have the computational cost C as an input parameter. Instead, the MLMC method has as input parameter ε, which determines how accurate the estimate is. We do not know beforehand what the computational cost of the MLMC method will be for a given value of ε. To ensure that each method uses the same computational cost in the simulation, we first estimate with the MLMC method for a given value of ε. The MLMC method will then return its estimator and the computational cost of its approximation of the expected shortfall. This computational cost will then be used as input parameters in the standard Monte Carlo and the adaptive sampling method.

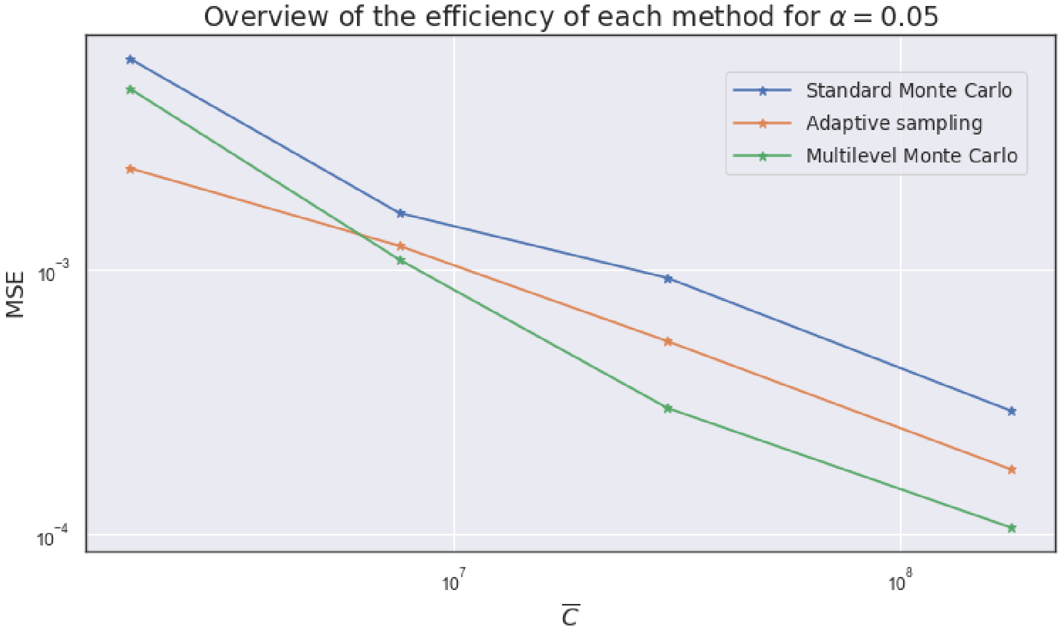
In our simulation process, we use the starting parameters stated in Table 7.1. We also approximate the expected shortfall for the confidence levels $\alpha \in \{0.05, 0.1, 0.2\}$. This should provide insight into whether the confidence level of the expected shortfall affects the efficiency of a method. For each confidence level, we shall use multiple values for ε when applying the MLMC method. This provides insight into how much the computational cost increases if a more accurate estimate has to be obtained. For ε holds that $\varepsilon \in \{0.1, 0.05, 0.0 25, 0.01\}$. These values are used as input parameters for the MLMC method, and the computational cost of the MLMC method is used as an input parameter for the standard Monte Carlo and the adaptive sampling method. Below are the other input parameters given for each method.

- **Multilevel Monte Carlo.** As already mentioned in section 7.4, the values M_0 and N_0 do depend on the level α, otherwise the expectation of $\hat{c}_{\alpha,0}$ cannot be computed. We have $M_0 = \alpha^{-1}$ and $N_0 = \lceil M_0/2 \rceil$, whereby $M_\ell = M_0 \cdot 2^{2\ell}$ and $N_\ell = N_0 \cdot 2^{2\ell}$. We set $G_0 = 10^3$, that is how many times the value $\hat{c}_{\alpha,0}$ is approximated to compute the value V_0, this is the same setting as in the numerical experiments of Giles [12]. After an approximation with the MLMC method, it does not return only its estimate of the expected shortfall but also its total computational cost C as defined in (6.11), which is used in the Standard Monte Carlo and adaptive sampling method.
- **Standard Monte Carlo.** Before this method is applied, we obtained the value C, which is equal to the computational cost of the MLMC method. For the standard Monte Carlo method, holds that $M = C^{2/3}$ and $N = C^{1/3}$.
- **Adaptive sampling.** As already stated in section 7.3, $L = 3$. The reason we do not select a higher value for L is that otherwise, the optimization problem in (5.21) becomes too complex. Here also holds that

$M = C^{2/3}$, where C equals the computational cost of the MLMC method. The value N_0 indicates how many inner stage samples are required to compute the simulated returns that are used to solve the optimization problem in (5.21). As the computational cost increases, we want to ensure that more inner stage samples are used to compute the row of simulated returns. This makes the row of simulated returns more accurate. Since the computational cost depends on ε, we set N_0 equal to ε^{-1}.

8.2. MSE comparison

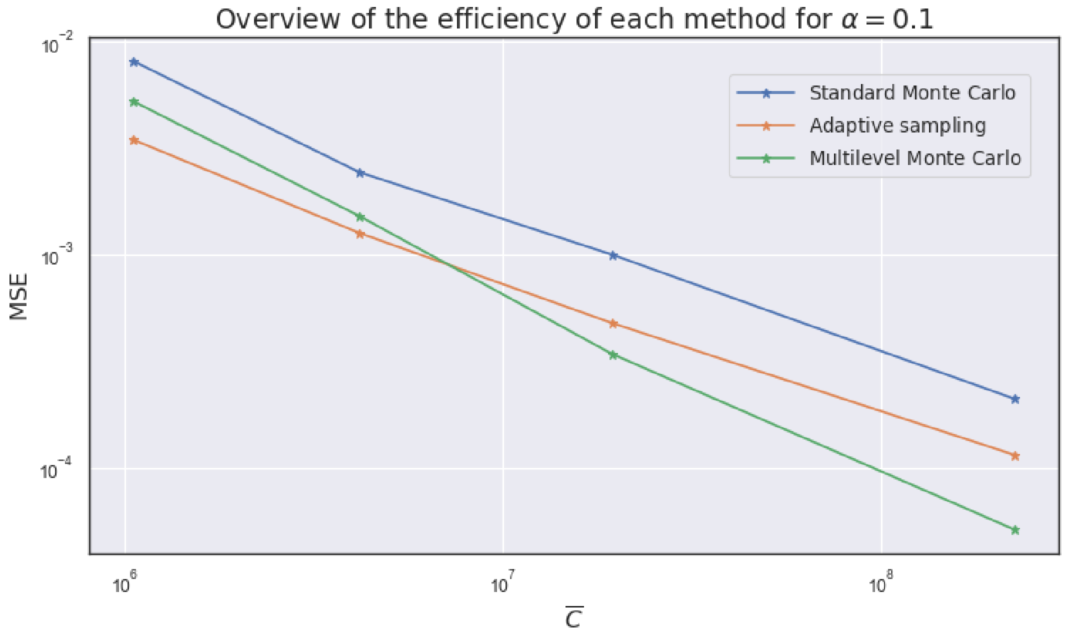
We want to determine which method is the most efficient for estimating the expected shortfall. We approximate the expected shortfall multiple times with each method, so we can compare their results. As mentioned before, we first use the MLMC method to approximate the expected shortfall and afterwards, we use its computational cost for the application of the standard Monte Carlo and adaptive sampling method. We denote $\overline{C}$ by the average computational cost that the MLMC method needed to approximate the expected shortfall for a given value of ε. The results are shown in Figure and Table 8.1 for $\alpha = 0.05$, Figure and Table 8.2 for $\alpha = 0.1$ and Figure and Table 8.3 for $\alpha = 0.2$.



ε	$\overline{C}$	Method	bias^2	Variance	MSE
0.1	1873068	Standard Monte Carlo	$3.462\cdot10^{-3}$	$2.774\cdot10^{-3}$	$6.237\cdot10^{-3}$
		Adaptive sampling	$1.986\cdot10^{-4}$	$2.211\cdot10^{-3}$	$2.410\cdot10^{-3}$
		Multilevel Monte Carlo	$3.417\cdot10^{-4}$	$4.472\cdot10^{-3}$	$4.816\cdot10^{-3}$
0.05	7533472	Standard Monte Carlo	$6.404\cdot10^{-4}$	$9.915\cdot10^{-4}$	$1.634\cdot10^{-3}$
		Adaptive sampling	$1.884\cdot10^{-4}$	$1.039\cdot10^{-3}$	$1.228\cdot10^{-3}$
		Multilevel Monte Carlo	$5.252\cdot10^{-5}$	$1.037\cdot10^{-3}$	$1.090\cdot10^{-3}$
0.025	29951216	Standard Monte Carlo	$4.895\cdot10^{-4}$	$3.621\cdot10^{-4}$	$8.517\cdot10^{-4}$
		Adaptive sampling	$1.332\cdot10^{-4}$	$4.036\cdot10^{-4}$	$5.368\cdot10^{-4}$
		Multilevel Monte Carlo	$4.958\cdot10^{-5}$	$2.510\cdot10^{-4}$	$3.006\cdot10^{-4}$
0.01	177668130	Standard Monte Carlo	$2.048\cdot10^{-4}$	$8.799\cdot10^{-5}$	$2.928\cdot10^{-4}$
		Adaptive sampling	$1.054\cdot10^{-4}$	$7.087\cdot10^{-5}$	$1.761\cdot10^{-4}$
		Multilevel Monte Carlo	$3.780\cdot10^{-5}$	$6.846\cdot10^{-5}$	$1.063\cdot10^{-4}$

Figure 8.1 & Table 8.1: Numerical results of multiple estimates of the standard Monte Carlo method, adaptive sampling method and the MLMC method when approximating the expected shortfall for $\alpha = 0.05$.

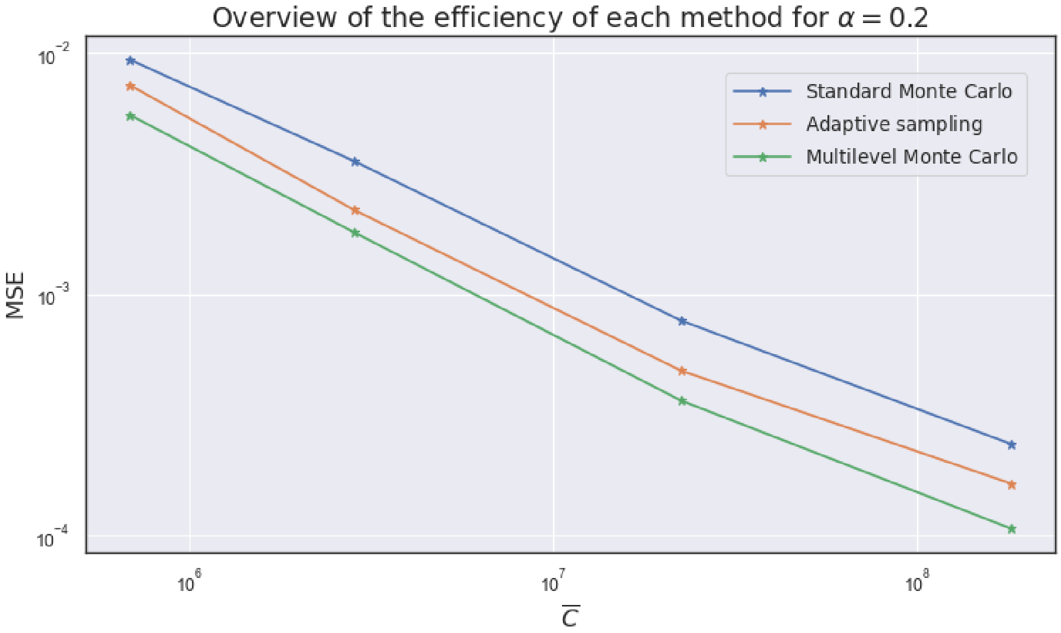
Figure and Table 8.1 show that it depends on the computational cost, which method has the lowest MSE for its estimators. For instance, for the lowest computational cost, adaptive sampling has the lowest MSE. However, when the computational cost increases, the MSE of the MLMC method decreases more significantly than that of the adaptive sampling method. So, with the MLMC method, the MSE decreases faster when the computational cost increases. Figure and Table 8.1 also show that the standard Monte Carlo method has the highest MSE values for all computational costs. Table 8.1 reveals a significant increase in the computational cost when the value of ε decreases.



ε	$\overline{C}$	Method	bias^2	Variance	MSE
0.1	1054191	Standard Monte Carlo	$4.664 \cdot 10^{-3}$	$3.409 \cdot 10^{-3}$	$8.074 \cdot 10^{-3}$
		Adaptive sampling	$1.999 \cdot 10^{-4}$	$3.255 \cdot 10^{-3}$	$3.455 \cdot 10^{-3}$
		Multilevel Monte Carlo	$9.992 \cdot 10^{-4}$	$4.249 \cdot 10^{-3}$	$5.237 \cdot 10^{-3}$
0.05	4185590	Standard Monte Carlo	$1.088 \cdot 10^{-3}$	$1.343 \cdot 10^{-3}$	$2.430 \cdot 10^{-3}$
		Adaptive sampling	$1.834 \cdot 10^{-5}$	$1.244 \cdot 10^{-3}$	$1.262 \cdot 10^{-3}$
		Multilevel Monte Carlo	$2.981 \cdot 10^{-4}$	$1.216 \cdot 10^{-3}$	$1.514 \cdot 10^{-3}$
0.025	19460903	Standard Monte Carlo	$5.083 \cdot 10^{-4}$	$4.926 \cdot 10^{-4}$	$1.001 \cdot 10^{-3}$
		Adaptive sampling	$5.671 \cdot 10^{-6}$	$4.724 \cdot 10^{-4}$	$4.781 \cdot 10^{-4}$
		Multilevel Monte Carlo	$7.577 \cdot 10^{-5}$	$2.668 \cdot 10^{-4}$	$3.426 \cdot 10^{-4}$
0.01	226590185	Standard Monte Carlo	$9.654 \cdot 10^{-5}$	$1.139 \cdot 10^{-4}$	$2.105 \cdot 10^{-4}$
		Adaptive sampling	$3.878 \cdot 10^{-5}$	$7.618 \cdot 10^{-5}$	$1.150 \cdot 10^{-4}$
		Multilevel Monte Carlo	$1.513 \cdot 10^{-6}$	$4.981 \cdot 10^{-5}$	$5.133 \cdot 10^{-5}$

Figure 8.2 & Table 8.2: Numerical results of multiple estimates of the standard Monte Carlo method, adaptive sampling method and the MLMC method when approximating the expected shortfall for $\alpha = 0.1$.

Figure and Table 8.2 illustrate that also for $\alpha = 0.1$, the standard Monte Carlo method has the highest MSE value for all computational costs. For a low computational cost, the MSE value for the adaptive sampling estimators is lower than the MLMC method estimators. However, the MSE value of the MLMC method also decreases more than adaptive sampling when the computational cost increases, so for a high computational cost, the MLMC method has the lowest MSE values.



ε	$\overline{C}$	Method	bias^2	Variance	MSE
0.1	685557	Standard Monte Carlo	$6.281\cdot10^{-3}$	$3.083\cdot10^{-3}$	$9.365\cdot10^{-3}$
		Adaptive sampling	$8.724\cdot10^{-4}$	$6.473\cdot10^{-3}$	$7.346\cdot10^{-3}$
		Multilevel Monte Carlo	$6.047\cdot10^{-4}$	$4.922\cdot10^{-3}$	$5.527\cdot10^{-3}$
0.05	2825099	Standard Monte Carlo	$2.156\cdot10^{-3}$	$1.409\cdot10^{-3}$	$3.565\cdot10^{-3}$
		Adaptive sampling	$9.338\cdot10^{-6}$	$2.228\cdot10^{-3}$	$2.237\cdot10^{-3}$
		Multilevel Monte Carlo	$7.766\cdot10^{-4}$	$1.031\cdot10^{-3}$	$1.808\cdot10^{-3}$
0.025	22409325	Standard Monte Carlo	$3.262\cdot10^{-4}$	$4.510\cdot10^{-4}$	$7.772\cdot10^{-4}$
		Adaptive sampling	$5.945\cdot10^{-5}$	$4.224\cdot10^{-4}$	$4.819\cdot10^{-4}$
		Multilevel Monte Carlo	$5.156\cdot10^{-5}$	$3.105\cdot10^{-4}$	$3.621\cdot10^{-4}$
0.01	181229184	Standard Monte Carlo	$1.413\cdot10^{-4}$	$9.763\cdot10^{-5}$	$2.390\cdot10^{-4}$
		Adaptive sampling	$3.423\cdot10^{-5}$	$1.296\cdot10^{-4}$	$1.638\cdot10^{-4}$
		Multilevel Monte Carlo	$3.115\cdot10^{-5}$	$7.550\cdot10^{-5}$	$1.066\cdot10^{-4}$

Figure 8.3 & Table 8.3: Numerical results of multiple estimates of the standard Monte Carlo method, adaptive sampling method and the MLMC method when approximating the expected shortfall for $\alpha = 0.2$.

In Figure and Table 8.3, the differences between the MSE values remain approximately constant. For a low value of the computational cost, the differences between the methods are slightly smaller than for a high computational cost. The MLMC method has the lowest MSE value for all computational costs, followed by the adaptive sampling method and the standard Monte Carlo method has the highest MSE value. For $\alpha = 0.05$, there is no variation in which method has the lowest MSE value for a given computational cost.

9

Conclusion & future research

In this section, we draw conclusions and support these with the results obtained from our numerical experiments. In addition, we will discuss potential follow-up research topics based on this thesis.

9.1. Conclusion

In this thesis, we examined how to apply the methods; standard Monte Carlo, adaptive sampling and the MLMC as efficiently as possible in a nested simulation framework setting. In addition, we compared the methods for the approximation of the expected shortfall. We draw our conclusions by analyzing the results obtained by the numerical experiments of this thesis.

For the standard Monte Carlo method, the main question was how the computational cost would be divided between the inner and outer stage samples. This had already been examined in the literature. Namely, the distribution can be determined by the product of a constant and a function. The only problem was that the constant was difficult to determine. In our numerical experiments, we observed that the constant could become negligible if the computational cost is large enough. So that the distribution between the inner and outer stage samples can be obtained quite easily for a given computational cost, when the given computational cost is high enough.

In the adaptive sampling method, we employed a technique that has already been developed and examined in the literature. However, we made some minor adjustments to this technique. For instance, this technique was applied to a process in which they only considered the loss of the portfolio. We applied to a process by which it could determine the expected shortfall based on the returns of the portfolio. In addition, this technique was applied to a small fixed number of risk scenarios. Our research has shown that this technique can also work very efficiently for a large number of risk scenarios in the nested simulation. The results in section 8 show that this method is more efficient than the standard Monte Carlo method. For a given computational cost, the estimators of the adaptive sampling method are always more accurate than those of the standard Monte Carlo method.

The application of the MLMC method for approximating the expected shortfall in the manner described throughout the thesis was, to our knowledge, not performed or examined before. This thesis has shown that the most efficient strategy for defining the value of M_ℓ and N_ℓ is as follows, $M_\ell = M_0 \cdot 2^{2\ell}$ and $N_\ell = N_0 \cdot 2^{2\ell}$. In addition, it is interesting to note that it does not make much difference to the efficiency of the method how the initial values M_0 and N_0 are defined. Figure 7.5 displays that the computational cost is not that far apart for the different values of α. This is certainly the case when the estimator has to be very accurate (i.e. for a low input value of ε). In that case, the computational cost for the three different values of α are very close to each other, and in this thesis, the initial values of M_0 and N_0 depend on the value α.

The main question of this thesis was to determine which method is most efficient for approximating the expected shortfall in a nested simulation. This study has shown that this method is the MLMC method. In section 8, we examined how accurate an estimator is and how much computational cost a method needs to obtain an estimate of the expected shortfall. The result was that the MLMC method produces the most accurate estimates for a high computational cost. Although, the results in section 8 also show that the MLMC method is not always the most efficient one. For instance, for $\alpha = 0.05$ and $\alpha = 0.1$, the adaptive sampling method provides the most accurate estimates for a lower computational cost. However, it is essential how much the computational cost increases when the estimator needs to be more accurate. The results show that the computational cost of the MLMC method increases less rapidly than the adaptive sampling method. Therefore, when a very accurate estimate has to be obtained and it has to be done in the fastest possible way, we recommend to apply the MLMC method.

9.2. Future research

Based on this thesis, there is the possibility of exploring new aspects of improving algorithms for approximating the expected shortfall in a nested simulation. Below, we have listed three potential follow-up studies, which we think could be of value to the academic world.

- Besides the MLMC method, the Multi-index Monte Carlo (MIMC) method is also applicable in a nested simulation. Haji-Ali, Nobile, and Tempone introduced this in 2016 [18]. In the MLMC method, the estimator's accuracy is determined by a 1-dimensional level ℓ. In the MIMC method the level ℓ has multiple dimensions, which thus determines the accuracy of the estimator. So the accuracy of the estimator can be influenced by several factors. Follow-up research could find out whether the MIMC method is even more efficient than the MLMC method. In this research, it should be taken into account that more factors should be included in the estimator's accuracy. For example, the number of time steps for discretizing a stochastic process can be included.
- Another interesting thing to examine is whether it is possible to combine the MLMC method with the adaptive sampling method for approximating the expected shortfall. Giles and Haji-Ali have done this before to approximate the Value-at-Risk [14, 15], but it required an additional stochastic root-finding algorithm. The combination of adaptive sampling and the MLMC method will be very similar to the underlying algorithm of the MLMC. The only difference is that there is no longer a constant growth for the value of N_ℓ when the value of ℓ increases. In that case, an adaptive sampling algorithm will be used to determine how much the value of N_ℓ will increase when the value of ℓ increases.
- In the adaptive sampling method, we have to solve an optimization problem to determine how the inner stage samples should be divided over the outer stage samples. Solving this optimization problem is quite complex and may take a lot of time. To make the adaptive sampling algorithm more accessible for estimating the expected shortfall, research is needed to find out how the distribution of the inner stage samples can be determined more quickly. One way to do this is to make the upper bound noted in proposition 5.3.3 even more simplified, allowing the optimization problem to proceed more quickly.

Bibliography

[1] Multilevel and multi-index Monte Carlo for portfolio value-at-risk. Oxford University MSc thesis, 2016.

[2] Carlo Acerbi and Dirk Tasche. On the coherence of expected shortfall. Journal of Banking & Finance, 26 (7):1487–1503, 2002.

[3] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. Mathematical finance, 9(3):203–228, 1999.

[4] Vijay S Bawa. Optimal rules for ordering uncertain prospects. Journal of Financial Economics, 2(1): 95–121, 1975.

[5] Bernard Bercu, Bernard Delyon, and Emmanuel Rio. Concentration inequalities for sums and martingales. Springer, 2015.

[6] Bruno Bouchard, Adil Reghai, and Benjamin Virrion. Computation of expected shortfall by fast detection of worst scenarios. Quantitative Finance, 21(7):1087–1108, 2021.

[7] Mark Broadie, Yiping Du, and Ciamac C Moallemi. Efficient risk estimation via nested sequential simulation. Management Science, 57(6):1172–1194, 2011.

[8] Mark Broadie, Yiping Du, and Ciamac C Moallemi. Risk estimation via regression. Operations Research, 63(5):1077–1097, 2015.

[9] Jessica O Dang, Ben M Feng, and Mary R Hardy. Efficient nested simulation of tail risk measures. In 2019 Winter Simulation Conference (WSC), pages 938–949. IEEE, 2019.

[10] Freddy Delbaen. Coherent risk measures on general probability spaces. In Advances in finance and stochastics, pages 1–37. Springer, 2002.

[11] Peter C Fishburn. Mean-risk analysis with risk associated with below-target returns. The American Economic Review, 67(2):116–126, 1977.

[12] Michael B Giles. Multilevel Monte Carlo path simulation. Operations Research, 56(3):607–617, 2008.

[13] Michael B Giles. Multilevel Monte Carlo methods. Acta Numerica, 24:259–328, 2015.

[14] Michael B Giles and Abdul-Lateef Haji-Ali. Multilevel nested simulation for efficient risk estimation. SIAM/ASA Journal on Uncertainty Quantification, 7(2):497–525, 2019.

[15] Michael B Giles and Abdul-Lateef Haji-Ali. Sub-sampling and other considerations for efficient risk estimation in large portfolios. arXiv preprint arXiv:1912.05484, 2019.

[16] Paul Glasserman. Monte Carlo Methods in Financial Engineering, volume 53. Springer, 2004.

[17] Allan D Grody, Kiran J Fernandes, Peter J Hughes, and J Steven Toms. Basel committee's fundamental review of the trading book: A commentary. Journal of Risk Management in Financial Institutions, 6(1): 6–9, 2013.

[18] Abdul-Lateef Haji-Ali, Fabio Nobile, and Raúl Tempone. Multi-index Monte Carlo: when sparsity meets sampling. Numerische Mathematik, 132(4):767–806, 2016.

[19] Abdul-Lateef Haji-Ali, Jonathan Spence, and Aretha Teckentrup. Adaptive multilevel Monte Carlo for probabilities. arXiv preprint arXiv:2107.09148, 2021.

[20] Desmond J Higham. An introduction to multilevel Monte Carlo for option valuation. International Journal of Computer Mathematics, 92(12):2347–2360, 2015.

[21] L Jeff Hong, Zhaolin Hu, and Guangwu Liu. Monte Carlo methods for value-at-risk and conditional value-at-risk: a review. ACM Transactions on Modeling and Computer Simulation (TOMACS), 24(4): 1–37, 2014.

[22] L Jeff Hong, Sandeep Juneja, and Guangwu Liu. Kernel smoothing for nested estimation with application to portfolio risk measurement. Operations Research, 65(3):657–673, 2017.

[23] Pavlo Krokhmal, Stanislav Uryasev, and Grigory Zrazhevsky. Risk management for hedge fund portfolios: a comparative analysis of linear rebalancing strategies. The Journal of Alternative Investments, 5(1):10–29, 2002.

[24] Pavlo Krokhmal, Michael Zabarankin, and Stan Uryasev. Modeling and optimization of risk. Handbook of the fundamentals of financial decision making: Part II, pages 555–600, 2013.

[25] Pavlo A Krokhmal. Higher moment coherent risk measures. Quantitative Finance, 7(4):373–387, 2007.

[26] Pierre L'Ecuyer and Art B Owen. Monte Carlo and Quasi-Monte Carlo Methods 2008. Springer, 2009.

[27] Shing-Hoi Lee and Peter W Glynn. Computing the distribution function of a conditional expectation via Monte Carlo: Discrete conditioning spaces. ACM Transactions on Modeling and Computer Simulation (TOMACS), 13(3):238–258, 2003.

[28] Christopher Lester, Christian Adam Yates, Michael B Giles, and Ruth E Baker. An adaptive multi-level simulation algorithm for stochastic biological systems. The Journal of chemical physics, 142(2): 01B612_1, 2015.

[29] Ming Liu, Barry L Nelson, and Jeremy Staum. An efficient simulation procedure for point estimation of expected shortfall. In Proceedings of the 2010 Winter Simulation Conference, pages 2821–2831. IEEE, 2010.

[30] BG Machael and J Sandeep. Nested simulation in portfolio risk management. Management Science, 56: 1833–1848, 2010.

[31] Siddhartha Mishra, Ch Schwab, and Jonas Šukys. Multi-level Monte Carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions. Journal of Computational Physics, 231(8): 3365–3388, 2012.

[32] Cornelis W Oosterlee, Lech A Grzelak, et al. Mathematical modeling and computation in finance: With exercises and python and matlab computer codes. World Scientific Books, 2019.

[33] Ludger Rüschendorf. Mathematical Risk Analysis. Springer Ser. Oper. Res. Financ. Eng. Springer, Heidelberg, 2013.

[34] Yue Xing, Tony Sit, and Hoi Ying Wong. Variance reduction for risk measures with importance sampling in nested simulation. Quantitative Finance, pages 1–17, 2021.

[35] Martin R Young. A minimax portfolio selection rule with linear programming solution. Management science, 44(5):673–683, 1998.

A

Proofs upper bound adaptive sampling method

The proofs of theorem 5.3.1 and proposition 5.3.3 have been copied from [6], and adapted to our simulation model.

Proof theorem 5.3.1

We have that:

$$\mathbb{E}\left[|\hat{c}_\alpha - c_\alpha|^p\right]^{\frac{1}{p}} \leq \underbrace{\mathbb{E}\left[\left|\frac{1}{\lfloor M\alpha \rfloor}\sum_{i\leq\lfloor M\alpha\rfloor} \hat{X}_{N_L}^{\mathbf{m}_{L-1}(\theta(\omega_i))} - X^{\mathbf{m}_{L-1}(\theta(\omega_i))}\right|^p\right]^{\frac{1}{p}}}_{\text{Monte Carlo error}} + \underbrace{\mathbb{E}\left[\left|\frac{1}{\lfloor M\alpha \rfloor}\sum_{i\leq\lfloor M\alpha\rfloor} X^{\mathbf{m}_{L-1}(\theta(\omega_i))} - X^{\theta(\omega_i)}\right|^p\right]^{\frac{1}{p}}}_{\text{permutation error}}$$

First we work out the permutation error. For the permutation error, we define $\mathcal{A}_{q,g}$ as the collection of subsets A of $[\![q+1, M]\!]$, such that $|A| = g$. Let then $S_q[A]$ be the collection of the q largest elements of the set A. Furthermore, we define $\mathcal{R}_\ell = J_\ell \backslash S_{q_\ell}[J_{\ell-1}]$. If we then have the set $\{i_1, ..., i_K\}$, we can define $\zeta(i_1) = \min \mathcal{R}_\ell$ and $\zeta(i_{k+1}) = \min\{j > \zeta(i_k) : j \in \mathcal{R}_\ell\}$. The permutation error can then be derived as follows:

$$\begin{aligned}
\mathbb{E}\left[\left|\frac{1}{\lfloor M\alpha \rfloor}\sum_{i\leq\lfloor M\alpha\rfloor} X^{\mathbf{m}_{L-1}(\theta(\omega_i))} - X^{\theta(\omega_i)}\right|^p\right]^{\frac{1}{p}} &\leq \frac{1}{\lfloor M\alpha \rfloor}\sum_{i\leq\lfloor M\alpha\rfloor}\sum_{\ell=1}^{L-1}\mathbb{E}\left[\left|X^{\theta(\omega_{\zeta(i)})} - X^{\theta(\omega_i)}\mathbf{1}_{\{i\in J_{\ell-1}\backslash J_\ell\}}\right|^p\right]^{\frac{1}{p}} \\
&\leq \max_{i\leq\lfloor M\alpha\rfloor}\sum_{\ell=1}^{L-1}\mathbb{E}\left[\left|X^{\theta(\omega_{\zeta(i)})} - X^{\theta(\omega_i)}\mathbf{1}_{\{i\in J_{\ell-1}\backslash J_\ell\}}\right|^p\right]^{\frac{1}{p}} \\
&\leq \max_{i\leq\lfloor M\alpha\rfloor}\sum_{\ell=1}^{L-1}\left(\max_{A\subset\mathcal{A}_{q_\ell,\delta q_\ell}}\sum_{k\in A}\mathbb{E}\left[\left|X^{\theta(\omega_k)} - X^{\theta(\omega_i)}\right|^p\mathbf{1}_{\{i\in J_{\ell-1}\backslash J_\ell,\zeta(i)=k\}}\right]\right)^{\frac{1}{p}} \\
&\leq \sum_{\ell=1}^{L-1}(\delta q_\ell)^{\frac{1}{p}}\max_{(i,k)\in[\![1,\lfloor\alpha M\rfloor]\!]\times[\![q_\ell+1,M]\!]}(X^{\theta(\omega_k)} - X^{\theta(\omega_i)})\mathbb{P}\left[\hat{X}_{N_\ell}^{\theta(\omega_k)} < \hat{X}_{N_\ell}^{\theta(\omega_i)}\right]^{\frac{1}{p}}
\end{aligned}$$

Now we are going to work out the corresponding Monte Carlo error:

$$\begin{aligned}
\mathbb{E}\left[\left|\frac{1}{\lfloor M\alpha \rfloor}\sum_{i\leq\lfloor M\alpha\rfloor} \hat{X}_{N_L}^{\mathbf{m}_{L-1}(\theta(\omega_i))} - X^{\mathbf{m}_{L-1}(\theta(\omega_i))}\right|^p\right]^{\frac{1}{p}} &\leq \frac{N_{L-1}}{N_L}\frac{1}{\lfloor M\alpha \rfloor}\mathbb{E}\left[\left|\sum_{i\leq\lfloor M\alpha\rfloor}\hat{X}_{N_{L-1}}^{\mathbf{m}_{L-1}(\theta(\omega_i))} - X^{\mathbf{m}_{L-1}(\theta(\omega_i))}\right|^p\right]^{\frac{1}{p}} \\
&+ \frac{N_L - N_{L-1}}{N_L}\frac{1}{\lfloor M\alpha \rfloor}\mathbb{E}\left[\left|\sum_{i\leq\lfloor M\alpha\rfloor}\frac{\sum_{j=N_{L-1}+1}^{N_L}\hat{X}_{j:N_L}^{\mathbf{m}_{L-1}(\theta(\omega_i))}}{N_L - N_{L-1}} - X^{\mathbf{m}_{L-1}(\theta(\omega_i))}\right|^p\right]^{\frac{1}{p}}
\end{aligned}$$

where

$$
\begin{aligned}
\mathbb{E}\left[\left|\sum_{i\leq\lfloor M\alpha\rfloor}\frac{\sum_{j=N_{L-1}+1}^{N_L}\hat{X}_{N_L:j}^{\mathbf{m}_{L-1}(\theta(\omega_i))}}{N_L-N_{L-1}}-X^{\mathbf{m}_{L-1}(\theta(\omega_i))}\right|^p\right]^{\frac{1}{p}} &= \mathbb{E}\left[\mathbb{E}\left[\left|\sum_{i\leq\lfloor M\alpha\rfloor}\delta\hat{X}_{N_L}^{\mathbf{m}_{L-1}(\theta(\omega_i))}-X^{\mathbf{m}_{L-1}(\theta(\omega_i))}\right|^p\middle|\mathbf{m}_{L-1}\right]\right]^{\frac{1}{p}}\\
&\leq\left(\max_{1\leq i_1<\ldots<i_{\lfloor\alpha M\rfloor}\leq M}\mathbb{E}\left[\left|\sum_{j\leq\lfloor M\alpha\rfloor}\delta\hat{X}_{N_L}^{\theta(\omega(i_j))}-X^{\theta(\omega(i_j))}\right|^p\right]\right)^{\frac{1}{p}}\\
&\leq\max_{1\leq i_1<\ldots<i_{\lfloor\alpha M\rfloor}\leq M}\sum_{j\leq\lfloor M\alpha\rfloor}\mathbb{E}\left[\left|\delta\hat{X}_{N_L}^{\theta(\omega(i_j))}-X^{\theta(\omega(i_j))}\right|^p\right]^{\frac{1}{p}}
\end{aligned}
$$

and

$$
\mathbb{E}\left[\left|\sum_{i\leq\lfloor M\alpha\rfloor}\hat{X}_{N_{L-1}}^{\mathbf{m}_{L-1}(\theta(\omega_i))}-X^{\mathbf{m}_{L-1}(\theta(\omega_i))}\right|^p\right]^{\frac{1}{p}}\leq\sum_{i\leq N}\mathbb{E}\left[\left|\hat{X}_{N_{L-1}}^{\theta(\omega_i)}-X^{\theta(\omega_i)}\right|^p\right]^{\frac{1}{p}}.
$$

□

Proof proposition 5.3.3

We start by showing that (5.15) is an upper bound of (5.4). With the use of theorem 5.3.2. we have that:

$$
\begin{aligned}
\mathbb{P}\left[\hat{X}_{N_\ell}^{\theta(\omega_i)}>\hat{X}_{N_\ell}^{\theta(\omega_k)}\right] &\leq \mathbb{P}\left[\hat{X}_{N_\ell}^{\theta(\omega_i)}\geq\hat{X}_{N_\ell}^{\theta(\omega_k)}\right]\\
&= \mathbb{P}\left[\hat{X}_{N_\ell}^{\theta(\omega_i)}-\hat{X}_{N_\ell}^{\theta(\omega_k)}\geq 0\right]\\
&= \mathbb{P}\left[\hat{X}_{N_\ell}^{\theta(\omega_i)}-\hat{X}_{N_\ell}^{\theta(\omega_k)}+X^{\theta(\omega_i)}-X^{\theta(\omega_i)}+X^{\theta(\omega_k)}-X^{\theta(\omega_k)}\geq 0\right]\\
&= \mathbb{P}\left[\left(\hat{X}_{N_\ell}^{\theta(\omega_i)}-X^{\theta(\omega_i)}\right)-\left(\hat{X}_{N_\ell}^{\theta(\omega_k)}-X^{\theta(\omega_k)}\right)\geq X^{\theta(\omega_k)}-X^{\theta(\omega_i)}\right]\\
&\leq \exp\left(-\frac{N_\ell\left(X^{\theta(\omega_k)}-X^{\theta(\omega_i)}\right)^2}{2\left(\sigma_{ik}^2+c\left(X^{\theta(\omega_k)}-X^{\theta(\omega_i)}\right)\right)}\right), \qquad \text{(A.1)}
\end{aligned}
$$

with $\sigma_{ik}^2=\mathbb{V}\left[\left(\hat{X}_{j:N}^{\theta(\omega_i)}-\hat{X}_{j:N}^{\theta(\omega_k)}\right)_{j=1}^{N}\right]$. Now we are going to show that (5.16) is an upper bound of (5.5):

$$
\begin{aligned}
\mathbb{E}\left[\left|\delta\hat{X}_{N_L}^{\theta(\omega_i)}-X^{\theta(\omega_i)}\right|^p\right] &= \int_0^\infty px^{p-1}\mathbb{P}\left[\left|\delta\hat{X}_{N_L}^{\theta(\omega_i)}-X^{\theta(\omega_i)}\right|\geq x\right]\mathrm{d}x\\
&\leq\int_0^\infty px^{p-1}e^{-\frac{\delta N_L x^2}{2\sigma_i^2+cx}}\,\mathrm{d}x\\
&\leq\int_0^\infty px^{p-1}e^{-\frac{\delta N_L x^2}{4\sigma_i^2}}\mathbf{1}_{\left\{x\leq\frac{\sigma_i^2}{c}\right\}}\mathrm{d}x+\int_0^\infty px^{p-1}e^{-\frac{\delta N_L x}{4c}}\mathbf{1}_{\left\{x>\frac{\sigma_i^2}{c}\right\}}\mathrm{d}x\\
&\leq\frac{p\sigma_i^p}{(\delta N_L)^{\frac{p}{2}}}\int_0^\infty y^{p-1}e^{-\frac{y^2}{4}}\mathrm{d}y+\frac{pc^p}{(\delta N_L)^p}\int_0^\infty y^{p-1}e^{-\frac{y}{4}}\mathrm{d}y\\
&\leq\frac{p\sigma_i^p}{(\delta N_L)^{\frac{p}{2}}}2^{p-1}\Gamma\left(\frac{p}{2}\right)+\frac{pc^p}{(\delta N_L)^p}4^p\Gamma(p) \qquad \text{(A.2)}
\end{aligned}
$$

Finally we show that (5.17) is an upper bound of (5.6):

$$\begin{aligned}
\mathbb{E}\left[\left|\delta\hat{X}_{N_{L-1}}^{\theta(\omega_i)} - X^{\theta(\omega_i)}\right|^p\right] &= \int_0^\infty p x^{p-1}\mathbb{P}\left[\left|\hat{X}_{N_{L-1}}^{\theta(\omega_i)} - X^{\theta(\omega_i)}\right| \geq x\right] \mathrm{d}x \\
&\leq \int_0^\infty p x^{p-1} e^{-\frac{N_{L-1}x^2}{2\sigma_i^2+cx}} \,\mathrm{d}x \\
&\leq \int_0^\infty p x^{p-1} e^{-\frac{N_{L-1}x^2}{4\sigma_i^2}} \mathbf{1}_{\left\{x\leq\frac{\sigma_i^2}{c}\right\}} \,\mathrm{d}x + \int_0^\infty p x^{p-1} e^{-\frac{N_{L-1}x}{4c}} \mathbf{1}_{\left\{x>\frac{\sigma_i^2}{c}\right\}} \,\mathrm{d}x \\
&\leq \frac{p\sigma_i^p}{(N_{L-1})^{\frac{p}{2}}} \int_0^\infty y^{p-1} e^{-\frac{y^2}{4}} \,\mathrm{d}y + \frac{pc^p}{(\delta N_{L-1})^p} \int_0^\infty y^{p-1} e^{-\frac{y}{4}} \,\mathrm{d}y \\
&\leq \frac{p\sigma_i^p}{(N_{L-1})^{\frac{p}{2}}} 2^{p-1}\Gamma\left(\frac{p}{2}\right) + \frac{pc^p}{(N_{L-1})^p} 4^p \Gamma(p). \qquad \text{(A.3)}
\end{aligned}$$

□

B
Demonstrating assumptions of the simulation

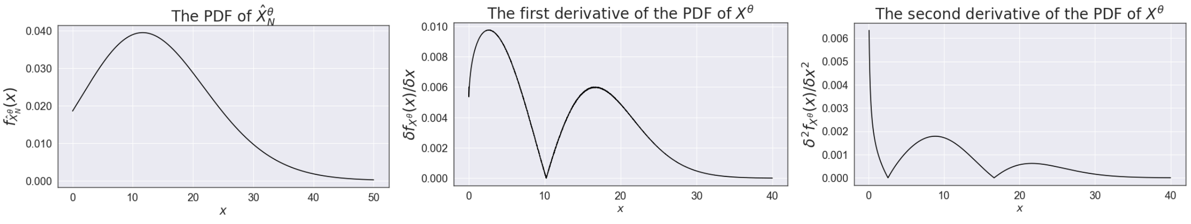
In section 4 and 5, we made assumptions of our simulation that helped us prove proposition 4.2.2 and proposition 5.3.3. We give an indication that these assumptions hold in our simulation process.

Assumption 4.2.1

For the standard Monte Carlo method, proposition 4.2.2 is used to determine the optimal ratio between the outer stage samples M and inner stage samples N. The proof of proposition 4.2.2 is made under the condition that assumption 4.2.1 holds. We give an indication that this assumption holds for this simulation. Assumption 4.2.1 consists of two items. The first item assumes that the joint PDF $g_N(\cdot,\cdot)$ of $\left(X^\theta, \hat{X}_N^\theta\right)$ and its partial derivatives $(\partial/\partial x)g_N(x,\hat{x})$ and $(\partial^2/\partial x^2)g_N(x,\hat{x})$ exists for each N and $(x,\hat{x})$. For $\hat{X}_N^\theta$ holds that as N increases, its PDF becomes increasingly similar to X^θ PDF. So we show that this assumption holds for $N=1$. We obtain the following CDF for $\hat{X}_N^\theta = e^{-r(T-\tau)}\max(S_T(\omega,W)-K,0)$, when $N=1$ and $x>0$,

$$F_{\hat{X}_N^\theta}(x) = \mathbb{P}\left(\hat{X}_N^\theta \leq x\right) = F_{\mathcal{N}(0,1)}\left(\frac{\log\frac{xe^{r-\tau}+K}{S_0} - \left(\mu-\sigma^2/2\right)\tau - \left(r-\sigma^2/2\right)(T-\tau)}{\sigma\sqrt{T}}\right). \tag{B.1}$$

Figure B.1a shows the PDF of $f_{\hat{X}_N^\theta}(x)$, where $N=1$ and $x>0$. Figure 7.1b already displays the PDF of X^θ, and Figures B.1b and B.1c show its derivatives $(\partial/\partial x)f_{X^\theta}(x)$ and $(\partial^2/\partial x^2)f_{X^\theta}(x)$. Therefore, the assumption that the joint PDF $g_N(x,\hat{x})$ and its partial derivatives $(\partial/\partial x)g_N(x,\hat{x})$ and $(\partial^2/\partial x^2)g_N(x,\hat{x})$ of $\left(X^\theta,\hat{X}_N^\theta\right)$ exists, can be drawn from Figure B.1.



(a) (b) (c)

Figure B.1: Illustration of the PDF of $\hat{X}_N^\theta$ in Figure B.1a, for $N=1$ and $x>0$, and the derivatives of the PDF of X^θ in Figures B.1b and B.1c.

For item 2 of Assumption 4.2.1, we have to show that the following holds,

$$\sup_N \int_{-\infty}^{\infty} |\hat{x}|^r \, p_{i,N}(\hat{x})\mathrm{d}\hat{x} < \infty \tag{B.2}$$

for all $i = 0,1,2$, $0 \le r \le 4$ and where,

$$g_N(x,\hat{x}) \le p_{0,N}(\hat{x}), \qquad \left|\frac{\partial}{\partial x} g_N(x,\hat{x})\right| \le p_{1,N}(\hat{x}), \qquad \left|\frac{\partial^2}{\partial x^2} g_N(x,\hat{x})\right| \le p_{2,N}(\hat{x}). \tag{B.3}$$

Since it always holds that $X^\theta \ge 0$, we only need to consider the integral from 0 to ∞. For $N \ge 1$ holds that,

$$g_N(x,\hat{x}) \le f_{\hat{X}_N^\theta}(\hat{x}), \qquad \left|\frac{\partial}{\partial x} g_N(x,\hat{x})\right| \le f_{\hat{X}_N^\theta}(\hat{x}), \qquad \left|\frac{\partial^2}{\partial x^2} g_N(x,\hat{x})\right| \le f_{\hat{X}_N^\theta}(\hat{x}). \tag{B.4}$$

We will show that $c \cdot \hat{x}^{-5}$ is an upper bound for $f_{\hat{X}_N^\theta}(\hat{x})$, where c is a constant. As N gets larger, the PDF of $\hat{X}_N^\theta$ becomes more and more similar to that of X^θ, where for $N \to \infty$ holds that both $\hat{X}_N^\theta$ and X^θ have the same PDF. Figure B.2 shows that the curve $c \cdot \hat{x}^{-5}$ is an upper bound for both the PDF of X^θ and $\hat{X}_N^\theta$, where $N = 1$. So then we have that,

$$\begin{aligned}
\sup_N \int_0^\infty |\hat{x}|^r\, p_{i,N}(\hat{x}) \mathrm{d}\hat{x} &\le \sup_N \int_0^\infty |\hat{x}|^r\, f_{\hat{X}_N^\theta}(\hat{x}) \mathrm{d}\hat{x} \\
&< \sup_N \left(\int_0^{50} |\hat{x}|^r\, f_{\hat{X}_N^\theta}(\hat{x}) \mathrm{d}\hat{x} + c \int_{50}^\infty |\hat{x}|^r\, \hat{x}^{-5} \mathrm{d}\hat{x} \right) \\
&< \infty
\end{aligned} \tag{B.5}$$

for all $i = 0,1,2$ and $0 \le r \le 4$. Since it holds that $\int_0^\infty \hat{x}^{-1} \mathrm{d}\hat{x} < \infty$.

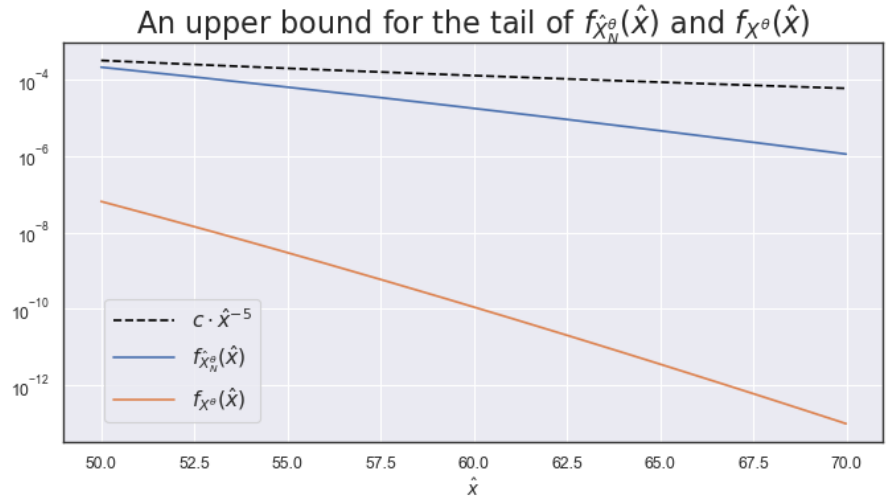


Figure B.2: Display of the upper bound $c \cdot \hat{x}^{-5}$, for the PDF of X^θ and $\hat{X}_N^\theta$, where $N = 1$.

Proposition 5.3.3

Proposition 5.3.3 assumes that there exists an $c \in \mathbb{R}_+$, such that the random variables $Z[i,k] = \left(\hat{X}_N^{\theta(\omega_i)} - X^{\theta(\omega_i)}\right) - \left(\hat{X}_N^{\theta(\omega_k)} - X^{\theta(\omega_k)}\right)$ are satisfying Bernstein's condition, for $i,k \le M$ and $d \ge 3$:

$$\mathbb{E}\left[|Z[i,k]|^d\right] \le \frac{d!c^{d-2}}{2} \mathbb{E}\left[Z[i,k]^2\right] \tag{B.6}$$

We have that $\lim_{N\to\infty} \hat{X}_N^{\theta(\omega)} = X^{\theta(\omega)}$. So the value $Z[i,k]$ gets smaller when N gets bigger. In our numerical experiments, every simulated return obtained by the adaptive sampling method has at least 10 inner stage samples. We are going to show that B.6 holds for $N = 10$.

Figure B.3 displays results of a simulation process where we computed the maximum of $Z[i,k]$, for $i,k \in$

$\{1,..,M\}$, with the simulated returns $\left(\hat{X}_N^{\theta(\omega_j)}\right)_{j=1}^{M}$. In the simulation is the maximum of $Z[i,k]$ computed for multiple values of M, where $M \in \{10^4, 5\cdot 10^4, 10^5, 5\cdot 10^5, 10^6\}$ and $i,k \in \{1,...,M\}$.

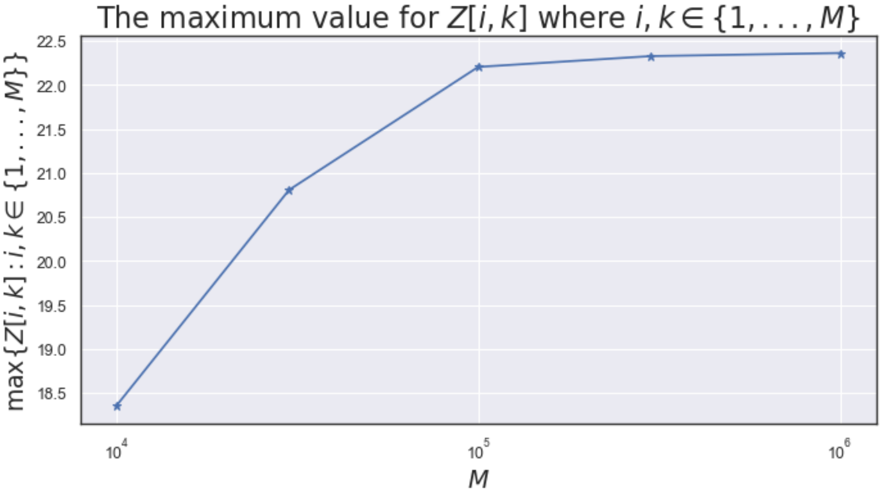


Figure B.3: Illustration of the maximum value of $Z[i,k]$ obtained from a simulation, where $i,k \in \{1,...,M\}$.

Figure B.3 shows that it does not make much difference to the maximum value of $Z[i,k]$ if the value M increases substantially. Therefore, we can conclude that there is always a $c \in \mathbb{R}_+$ for which holds that,

$$\mathbb{E}\left[|Z[i,k]|^d\right] \leq \frac{d!c^{d-2}}{2}\mathbb{E}\left[Z[i,k]^2\right] \tag{B.7}$$

for $i,k \in \{1,...,M\}$ and $d \geq 3$.

C Python code

Packages

```
import math
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
from matplotlib.pyplot import figure
from scipy import optimize
from scipy.optimize import minimize_scalar
from scipy.integrate import quad
```

Starting parameters

```
S0 = 100
K = 90
sigma = 0.20
T = 0.25
tau = 0.10
r = 0.07
mu = 0.04
```

Analytical computation of the expected shortfall

```
def BS_option_price(Stau,K,sigma,T,tau,r):
    T_tau = T - tau
    d1 = (np.log(Stau / K) + (r + 0.5 * np.power(sigma,2.0)) * T_tau) / float(sigma * np.sqrt(T_tau))
    d2 = d1 - sigma * np.sqrt(T_tau)
    value = Stau*norm.cdf(d1) - K * np.exp(-r * T_tau) * norm.cdf(d2)
    return value

def q(x, S0, K, sigma, T, tau, r, mu):
    w = norm.ppf(x,0,1)
    Stau_alpha = S0*math.exp((mu - 0.5*sigma**2)*tau + sigma*math.sqrt(tau)*w)
    option_T = BS_option_price(Stau_alpha,K,sigma,T,tau,r)
    return option_T

I = quad(q,0,alpha,args=(S0, K, sigma, T, tau, r, mu))
ES = (-1/alpha)*I[0]
```

Generating simulated returns

```
def generate_outer_sample(S0,tau,mu,sigma):
    w = np.random.normal(0,1)
    outer_sample = S0*math.exp((mu - 0.5*sigma**2)*tau + sigma*math.sqrt(tau)*w)
    return outer_sample

def generate_return(Stau, N, T, tau, r, sigma, K):
    T_tau = T - tau
    lst_W = np.random.normal(0,1,N)
    ST = Stau*np.exp((r - 0.5*sigma**2)*T_tau + sigma*np.sqrt(T_tau)*lst_W)
    lst_returns = np.exp(-r*(T_tau))*np.maximum(ST - K, 0)
    return lst_returns
```

Standard Monte Carlo method

```
def StandardMC(M,N,alpha,S0,K,sigma,T,tau,r,mu):
    lst_returns_i = []
    for i in range(M):
        Stau = generate_outer_sample(S0,tau,mu,sigma)
        lst_returns_conditionali_j = generate_return(Stau, N, T, tau, r, sigma, K)
        lst_returns_i.append(np.mean(lst_returns_conditionali_j))
    lst_returns_i.sort()
    M_alpha = math.floor(M*alpha)
    expected_shortfall = (-1.0/M_alpha)*sum(lst_returns_i[0:M_alpha])
    return expected_shortfall
```

Adaptive sampling

```
def AdaptiveMC(M,N0,alpha,L,C,S0,K,sigma,T,tau,r,mu):
    lst_return__j_l__inner_samples = []
    for i in range(M):
        Stau = generate_outer_sample(S0,tau,mu,sigma)
        lst_return__j_l__inner_samples.append([0,Stau,0])
        lst_returns_conditionali_j = generate_return(Stau, N0, T, tau, r, sigma, K)
        lst_return__j_l__inner_samples[i][2] = lst_returns_conditionali_j
        lst_return__j_l__inner_samples[i][0] = np.mean(lst_return__j_l__inner_samples[i][2])
    opt_q_N = optimal_q_N(lst_return__j_l__inner_samples, alpha, C, N0)
    set_N = opt_q_N['optimal_N']
    set_q = opt_q_N['optimal_q']
    M = set_q[0]
    M_alpha = set_q[2]
    lst_return__j_l__inner_samples = lst_return__j_l__inner_samples[0:M]
    for l in range(L):
        for i in range(len(lst_return__j_l__inner_samples)):
            Stau = lst_return__j_l__inner_samples[i][1]
            if l == 0:
                lst_returns_conditionali_j = generate_return(
                            Stau, set_N[l] - N0, T, tau, r, sigma, K)
                lst_return__j_l__inner_samples[i][2] = np.concatenate((
                            lst_return__j_l__inner_samples[i][2],lst_returns_conditionali_j))
                lst_return__j_l__inner_samples[i][0] = np.mean(
                            lst_return__j_l__inner_samples[i][2])
            if l != 0:
                lst_returns_conditionali_j =  generate_return(
                            Stau, set_N[l]-set_N[l-1], T, tau, r, sigma, K)
                lst_return__j_l__inner_samples[i][2] = np.concatenate((
                            lst_return__j_l__inner_samples[i][2],lst_returns_conditionali_j))
                lst_return__j_l__inner_samples[i][0] = np.mean(lst_return__j_l__inner_samples[i][2])
        lst_return__j_l__inner_samples.sort()
        if l != L-1:
            lst_return__j_l__inner_samples = lst_return__j_l__inner_samples[0:set_q[l+1]]
    lst_lowest_returns = []
    for i in range(M_alpha):
        lst_lowest_returns.append(lst_return__j_l__inner_samples[i][0])
    expected_shortall = -1*np.mean(lst_lowest_returns)
    result_adap = {'ES_adap':expected_shortall, 'set_N':set_N, 'set_q':set_q}
    return result_adap


def optimal_q_N(lst_return__j_l__inner_samples, alpha, C, N0):
    lst_return__j_l__inner_samples_1 = objective_list(lst_return__j_l__inner_samples)
    M_1 = len(lst_return__j_l__inner_samples_1)
    C_1 =  round(C**(1/3)*M_1)
    ratio_q = math.floor(C**(2/3)/M_1)
    M_alpha = math.floor(M_1*alpha)
    smallest_error = math.inf
    for q1 in np.arange(M_alpha + 1,M_1,2):
        q0 = M_1
        q2 = M_alpha
        q = [q0,q1,q2]
        N = optimum_N(q,M_1,alpha,N0,C_1,lst_return__j_l__inner_samples_1)
        error = F_2(q,N,lst_return__j_l__inner_samples_1,M_1,alpha)
        if error < smallest_error:
            optimal_N = N
            optimal_q = [q0*ratio_q,q1*ratio_q,q2*ratio_q]
            smallest_error = error
    return {'optimal_q':optimal_q,'optimal_N':optimal_N}

def objective_list(lst):
    M = len(lst)
    lst.sort()
    small_list = []
    for i in range(1,21):
        index_lst = round(i*(M/20.0)) - 1
        small_list.append(lst[index_lst])
    return obj_list

def optimum_N(q,M,alpha,N0,C,lst_return__j_l__inner_samples):
    upper_bound_N1 = math.floor(C/q[0])
    N1 = round(minimize_scalar(objective_function_2,
                               method = 'bounded',bounds = (N0,upper_bound_N1),
                               args=(q,lst_return__j_l__inner_samples,M,alpha,C))['x'])
    upper_bound_N2 = math.floor((C + N1*(q[1] - q[0]))/q[1]) - 1
    N2 = round(minimize_scalar(objective_function_1,
                               method = 'bounded',bounds = (N1,upper_bound_N2),
                               args=(N1,q,lst_return__j_l__inner_samples,M,alpha,C))['x'])
    N3 = options_N3(q,N1,N2,C)
    return [int(N1),int(N2),int(N3)]

def objective_function_1(N2,N1,q,lst_return__j_l__inner_samples,M,alpha,C):
    N3 = options_N3(q,N1,N2,C)
    return F_2(q,[N1,N2,N3],lst_return__j_l__inner_samples,M,alpha)
```

```
def objective_function_2(N1,q,lst_return__j_l__inner_samples,M,alpha,C):
    upper_bound_N2 = math.floor((C + N1*(q[1] - q[0]))/q[1]) - 1
    N2 = minimize_scalar(objective_function_1,
                         method = 'bounded',bounds = (N1,upper_bound_N2),
                         args=(N1,q,lst_return__j_l__inner_samples,M,alpha,C))['x']
    N3 = options_N3(q,N1,N2,C)
    return F_2(q,[N1,N2,N3],lst_return__j_l__inner_samples,M,alpha)

def options_N3(q,N1,N2,C):
    return  math.floor((C - q[0]*N1 - q[1]*(N2 - N1))/q[2] + N2)

def F_2(q,N,lst_return__j_l__inner_samples,M,alpha):
    lst_return__j_l__inner_samples.sort()
    M_alpha = math.floor(M*alpha)
    L = len(q)
    lst_return_i = []
    lst_variance_i = []
    matrix_variance_i_k = []
    for i in range(M):
        lst_return_i.append(lst_return__j_l__inner_samples[i][0])
        inner_samples_i = np.array(lst_return__j_l__inner_samples[i][2])
        lst_variance_i.append(np.var(inner_samples_i))
        lst_variance_i_k = []
        for k in range(len(lst_return__j_l__inner_samples)):
            inner_samples_k = np.array(lst_return__j_l__inner_samples[k][2])
            lst_variance_i_k.append(np.var(inner_samples_i  - inner_samples_k))
        matrix_variance_i_k.append(lst_variance_i_k)
    sum_first_term = 0
    for l in range(1,L):
        delta_q = q[l-1] - q[l]
        lst_errorbound_1 = []
        for i in range(M_alpha):
            for k in range(q[l],M):
                return_k = lst_return_i[k]
                return_i = lst_return_i[i]
                lst_errorbound_1.append((return_k - return_i)*
                                        math.exp(-(N[l-1]*(return_k - return_i)**2)/
                                                 (4*matrix_variance_i_k[i][k])))
        sum_first_term = sum_first_term + delta_q**(0.5)*(1.0/alpha)*max(lst_errorbound_1)
    lst_errorbound_2 = []
    for i in range(M):
        delta_NL = N[-1] - N[-2]
        p_sigma_ij_p = 2*(lst_variance_i[i]**2)
        lst_errorbound_2.append(C_p(2)*(p_sigma_ij_p/(delta_NL)))
    lst_errorbound_2.sort()
    if lst_errorbound_2[M-1] == math.inf:
        sum_second_term = math.inf
    else:
        sum_second_term = (1.0/M_alpha)*(delta_NL/N[-1])*sum(
                                            lst_errorbound_2[M - M_alpha:M])**(0.5)
    lst_errorbound_3 = []
    NL_1 = N[-2]
    for i in range(M):
        p_sigma_ij_p = 2 * (lst_variance_i[i]**2)
        lst_errorbound_3.append(C_p(2)*(p_sigma_ij_p/(NL_1)))
    sum_third_term = (1.0/M_alpha)*(N[-2]/N[-1])*sum(lst_errorbound_3)**(0.5)
    return sum_first_term + sum_second_term + sum_third_term

def C_p(p):
    return 2**(p-1)*math.gamma(p/2.0)
```

Multilevel Monte Carlo method

```
def MLMC(M0,N0,G0,alpha,eps,S0,K,sigma,T,tau,r,mu):
    L = 0
    G = G0
    Y = 'not converged'
    lst_Yl_old_l = []
    lst_Vl_old_l = []
    lst_Cl_l = []
    lst_ES_l_Gl = []
    lst_Gl_old_l = []
    while Y == 'not converged':
        lst_ES_G = []
        for g in range(G):
            if L == 0:
                expected_shortfall_g = StandardMC(M0,N0,alpha,S0,K,sigma,T,tau,r,mu)
            if L != 0:
                expected_shortfall_g = Antithetic(M_l(M0,L),N_l(N0,L),
                                                 M_l(M0,L-1),N_l(N0,L-1),
                                                 alpha,S0,K,sigma,T,tau,r,mu)
            lst_ES_G.append(expected_shortfall_g)
        VL = np.var(lst_ES_G)
        if L == 0:
            CL = M0*N0
        if L != 0:
            CL = M_l(M0,L)*(N_l(N0,L) + N_l(N0,L-1))
        lst_Vl_old_l.append(VL)
        lst_Cl_l.append(CL)
        lst_Yl_new_l = []
        lst_Vl_new_l = []
        lst_Gl_new_l = []
        lst_Gl_old_l.append(0)
        lst_ES_l_Gl.append([])
        for l in range (L+1):
            Vl = lst_Vl_old_l[l]
            Cl = lst_Cl_l[l]
            Gl_new =  math.ceil(2*eps**(-2)*math.sqrt(Vl*Cl**(-1))*
                                sum([math.sqrt(a*b) for a, b in zip(lst_Vl_old_l, lst_Cl_l)]))
            Gl_old = lst_Gl_old_l[l]
            lst_Gl_new_l.append(Gl_new)
            if Gl_old < Gl_new:
                for g in range(int(Gl_new - Gl_old)):
                    if l == 0:
                        expected_shortfall_g = StandardMC(M0,N0,alpha,S0,K,sigma,T,tau,r,mu)
                    if l != 0:
                        expected_shortfall_g = Antithetic(M_l(M0,l),N_l(N0,l),
                                                         M_l(M0,l-1),N_l(N0,l-1),
                                                         alpha,S0,K,sigma,T,tau,r,mu)
                    lst_ES_l_Gl[l].append(expected_shortfall_g)
            if lst_ES_l_Gl[l] == []:
                Yl = 0
                Vl = 0
            else:
                Yl = np.mean(lst_ES_l_Gl[l])
                Vl = np.var(lst_ES_l_Gl[l])
            lst_Yl_new_l.append(Yl)
            lst_Vl_new_l.append(Vl)
        G = Gl_new
        lst_Yl_old_l = lst_Yl_new_l
        lst_Vl_old_l = lst_Vl_new_l
        lst_Gl_old_l = lst_Gl_new_l
        S = int((1.0*M_l(M0,L))/M_l(M0,L-1))
        if L >= 2 and max(S**(-1)*abs(lst_Yl_new_l[L-1]),
                          abs(lst_Yl_new_l[L])) < (1/math.sqrt(2))*(S-1)*eps:
            Y = 'converged'
            C = sum([a*b for a, b in zip(lst_Cl_l, lst_Gl_new_l)])
            expected_shortfall = sum(lst_Yl_new_l)
            Result_MLMC = {'ES_MLMC':expected_shortfall,'C_MLMC':C,
                           'lst_Yl':lst_Yl_old_l,'lst_Vl': lst_Vl_old_l,'lst_Gl': lst_Gl_old_l,}
        L = L + 1
    return  Result_MLMC

def Antithetic(ML,NL,Ml,Nl,alpha,S0,K,sigma,T,tau,r,mu):
    S = int((1.0*ML)/Ml)
    Ml_alpha = math.floor(Ml*alpha)
    ML_alpha = math.floor(ML*alpha)
    lst_expected_shorfall_s = []
    lst_returns_si = []
    for s in range(S):
        lst_returns_i = []
        for i in range(Ml):
            Stau = generate_outer_sample(S0,tau,mu,sigma)
            lst_returns_conditionalsi_j = generate_return(Stau, NL, T, tau, r, sigma, K)
            lst_returns_i.append(np.mean(lst_returns_conditionalsi_j[0:Nl]))
            lst_returns_si.append(np.mean(lst_returns_conditionalsi_j[0:NL]))
        lst_returns_i.sort()
        expected_shortfall_s = (-1.0/Ml_alpha)*sum(lst_returns_i[0:Ml_alpha])
        lst_expected_shorfall_s.append(expected_shortfall_s)
    lst_returns_si.sort()
    expected_shortfall = (-1.0/ML_alpha)*sum(lst_returns_si[0:ML_alpha])
    result = expected_shortfall - np.mean(lst_expected_shorfall_s)
    return result

def M_l(M0,l):
    return M0*(2**(2*l))

def N_l(N0,l):
    return N0*(2**(2*l))
```