Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Master Thesis

Computation of Risk Measures using Importance Sampling

Philippe Muller

Supervisors:
Prof. Dr. Paul Embrechts (ETH Zürich)
Assoc. Prof. Dr. Henrik Hult (KTH)
Assoc. Prof. Dr. Filip Lindskog (KTH)

Stockholm, February 2010

Abstract

Estimation of Value-at-Risk and expected shortfall using standard Monte Carlo can result in high computational cost. We make a review of importance sampling, a common method to make estimations more efficient. A "direct" approach to compute risk measures from simulations drawn from an importance sampling density is described in detail. We explain how to select an efficient importance sampling distribution for loss probability estimations in the case of normally distributed risk factor changes. Some algorithms for efficient risk measure computations are presented explicitly. By considering numerical examples, we analyze the effect of regularly updating the importance sampling density during the simulation process.

1 Introduction

For a holder of a financial portfolio it is important to estimate the risk of his position. To quantify risk, or uncertainty, we use so-called risk measures. Let us assume the following situation to be able to clearly define the two most frequently used risk measures, Value-at-Risk and expected shortfall. We want to study the loss $L = -(V_t - V_0)$ of a given portfolio during a time period of length t. V_0 and V_t are supposed to be the initial portfolio value and the portfolio value after time period t, respectively. Positive values of L correspond to losses whereas negative values of L are considered to be gains. Let F_L be the distribution function of L. Let us now define Value-at-Risk.

Definition 1.0.1 *Value-at-Risk* (VaR) at level $\alpha \in (0, 1)$ of our portfolio is defined by

$$\mathrm{VaR}_\alpha = \min\{m : P(L \leq m) \geq 1 - \alpha\},$$

that is, the smallest value m such that the probability of the portfolio loss $L = -(V_t - V_0)$ being at most m is at least $1 - \alpha$.

Remark 1.0.2 In general VaR_α of a portfolio is referred to as the smallest amount of money that, if added to the portfolio at time 0 and invested in a risk-free asset, ensures that the probability of a strictly negative final net worth at time t is not greater than α. In other words, VaR_α corresponds in general to the smallest value m such that the probability of the discounted portfolio loss $L = -(V_1 \cdot e^{-rt} - V_0)$ being at most m is at least $1 - \alpha$, where e^{-rt} is the discount factor corresponding to the continuously compounded risk-free interest rate r. In Definition 1.0.1 we consider a simplified version of VaR, based on the portfolio loss instead of the discounted portfolio loss. This allows us to keep things simpler and for small holding periods t there is no big difference in the results.

Since $\min\{m : P(L \leq m) \geq 1-\alpha\} = \min\{m : F_L(m) \geq 1-\alpha\} = F_L^{-1}(1-\alpha)$, we can rewrite VaR_α of our portfolio as

$$\mathrm{VaR}_\alpha = F_L^{-1}(1 - \alpha). \tag{1}$$

That is, VaR can be interpreted as a quantile of the loss distribution. The most common values for the parameter α are 1% and 5%, corresponding to the 99% and the 95% quantile of the loss distribution function F_L. VaR is the most frequently used risk measure, but it has some drawbacks. Value-at-Risk is not subadditive, that is, it does not reward diversification. A second limitation is that there is no information about how big losses may be in case the threshold VaR_α is exceeded. Let us now define expected shortfall, which

does not have these shortcomings. We consider the same portfolio as for the definition of VaR.

Definition 1.0.3 *Expected shortfall* (ES) at level $\alpha \in (0,1)$ is the average Value-at-Risk for levels $p \leq \alpha$, that is

$$\mathrm{ES}_\alpha = \frac{1}{\alpha} \int_0^\alpha \mathrm{VaR}_p \, dp.$$

By (1) follows that, for an investment period from 0 to t, expected shortfall for a financial portfolio can be written as

$$\mathrm{ES}_\alpha = \frac{1}{\alpha} \int_0^\alpha F_L^{-1}(1-p)dp = \frac{1}{\alpha} \int_{1-\alpha}^1 F_L^{-1}(p)dp. \tag{2}$$

Exact formulas for VaR and ES are not available in most models. That is why usually Monte Carlo simulation is used to compute risk measure estimations.

Using standard Monte Carlo to estimate loss probabilities $P(L > x)$ for large loss thresholds x can be very inefficient. The same applies to extreme quantiles. The Monte Carlo estimate for $p = P(L > x)$ is given by

$$\hat{p} = \frac{1}{N} \sum_{i=1}^{N} I\{L_i > x\},$$

where $L_1, L_2, \ldots, L_N$ are considered to be N portfolio loss simulations. For large values of x, a lot of simulations are needed to get accurate estimations, because $I\{L_i > x\} = 0$ in most of the cases. Hence, standard Monte Carlo may result in a lot of computational effort, especially for heavy tailed loss distributions. Another way of showing how inefficient standard Monte Carlo can be is to have a look at

$$\mathrm{std}(\hat{p}) = \frac{1}{\sqrt{N}} \cdot \sqrt{p(1-p)} \overset{\text{(for } p \text{ small)}}{\approx} \frac{\sqrt{p}}{\sqrt{N}},$$

the standard deviation of the above estimator. To avoid too much variation in the sample, $\mathrm{std}(\hat{p}) \approx p$ should be satisfied and hence $N \approx \frac{1}{p}$. This can result in a huge number of simulations and an inefficient estimator. We would like to make sampling more efficient. This can be done by importance sampling (IS), which is a variance reduction technique and hence reduces the required number of simulations.

This thesis is organized as follows. In Section 2 we review the theory on importance sampling and introduce the notion of exponential twisting. In

Section 3 we describe a "direct" approach to estimate risk measures using Monte Carlo simulation and importance sampling. In Section 4 we show how to use first and second order Taylor approximations of L to make loss probability estimations more efficient through importance sampling. We restrict ourselves to the light tailed case. In Section 5 some algorithms for efficient VaR and ES estimations are introduced. Finally, in Section 6 we apply the algorithms to specific portfolios and compare their performances.

2 Importance sampling

2.1 Principles of importance sampling

In [Glasserman 2003] the theoretical background of importance sampling has been well presented. Here we recall the main points and try to further illustrate them. Let X be an $\mathbb{R}^d$-valued random variable with probability density f. h is supposed to be a function from $\mathbb{R}^d$ to $\mathbb{R}$. We want to estimate $\alpha = \mathrm{E}[h(X)] = \int h(x)f(x)dx$. The standard Monte Carlo estimator is given by

$$\hat{\alpha} = \frac{1}{N}\sum_{i=1}^{N} h(X_i), \tag{3}$$

where $X_1, X_2, \ldots, X_N$ are N independent simulations drawn from f. As already mentioned in the introduction, this estimator can be very inefficient. Let us try to increase sampling efficiency by giving more weight to "important" outcomes, using a change of measure. For instance, large losses are considered to be "important" for estimating extreme loss probabilities. Instead of using f for simulating, we will now draw $X_1, X_2, \ldots, X_N$ from g, a new probability density on $\mathbb{R}^d$, called *importance sampling density*. The density g has to satisfy

$$f(x) > 0 \Rightarrow g(x) > 0 \ \forall x \in \mathbb{R}^d.$$

Of course we cannot simply apply (3) to the new X_i to estimate α, since this would lead to a biased estimator. Instead we use

$$\hat{\alpha}_g = \frac{1}{N}\sum_{i=1}^{N} h(X_i)\frac{f(X_i)}{g(X_i)} \tag{4}$$

as the importance sampling estimator associated with g. For each X_i we use $f(X_i)/g(X_i)$, the *likelihood ratio* or *Radon-Nikodym derivative*, for compensation. Since

$$\tilde{\mathrm{E}}[\hat{\alpha}_g] = \tilde{\mathrm{E}}\left[\, h(X_i)\frac{f(X_i)}{g(X_i)} \,\right] = \int h(X)\frac{f(X)}{g(X)}g(X)dx = \int h(X)f(X)dx = \alpha,$$

we can conclude that $\hat{\alpha}_g$ is an unbiased estimator of α. $\tilde{\mathrm{E}}[\cdot]$ denotes expectation when g has been used for sampling.

Remark 2.1.1 We will see later that, instead of using the same importance sampling density for all the simulations, it may be reasonable to use g_i to draw X_i, for $i = 1, 2, \ldots, N$. Let us therefore mention at this stage that

$$\hat{\alpha}_{g_1,g_2,\ldots,g_N} = \frac{1}{N} \sum_{i=1}^{N} h(X_i) \frac{f(X_i)}{g_i(X_i)} \tag{5}$$

is also an unbiased estimator for α. Using $\hat{\alpha}_{g_1,g_2,\ldots,g_N}$ as an estimator offers the possibility to update the importance sampling density during the process of simulating $X_1, X_2, \ldots, X_N$.

The goal of importance sampling is variance reduction of the estimator $\hat{\alpha}$. Since

$$\mathrm{E}[\hat{\alpha}] = \tilde{\mathrm{E}}[\hat{\alpha}_g] = \alpha,$$

a comparison of variances with and without importance sampling can be done by comparing second moments of $\hat{\alpha}_g$ and $\hat{\alpha}$. The second moment of $\hat{\alpha}_g$ is given by

$$\tilde{\mathrm{E}}\left[\, \hat{\alpha}_g^2 \,\right] = \tilde{\mathrm{E}}\left[\left(h(X) \frac{f(X)}{g(X)} \right)^2 \right] = \mathrm{E}\left[h(X)^2 \frac{f(X)}{g(X)} \right]$$

and will be compared to

$$\mathrm{E}\left[\, \hat{\alpha}^2 \,\right] = \mathrm{E}\left[\, h(X)^2 \,\right] .$$

Depending on how g has been chosen, the variance of $\hat{\alpha}_g$ may become much larger or much smaller than the variance of $\hat{\alpha}$. That is, the success of importance sampling directly depends on the choice of the density g. But how should g be chosen?

If, as in this thesis, we want to estimate probabilities, h is an indicator function and hence is nonnegative. That is, without any restriction we can now consider the special case in which $h \geq 0 \; \forall x \in \mathbb{R}^d$. In this case $h(x)f(x) \geq 0 \; \forall x \in \mathbb{R}^d$ and we may therefore normalize $h(x)f(x)$ to a probability density. Let g be this density. Then $h(X)f(X)/g(X)$ is constant, resulting in a perfect estimator $\hat{\alpha}_g$ with zero variance. This special case is of course not applicable in practice, because for the normalization of the product $h(x)f(x)$ we need to divide it by α itself, which is generally unknown. But at least we can use this information to get some guidance in choosing

an effective g. The above special case tells us that we should always try to sample in proportion of the product $h \cdot f$.

It is worth noting that, although

$$\tilde{\mathrm{E}}\left[\frac{f(X_i)}{g(X_i)}\right] = \tilde{\mathrm{E}}\left[\frac{f(X)}{g(X)}\right] = \int f(X)dx = 1$$

regardless of the value of X_i, likelihood ratios can take huge values with small but non-negligible probability. This in turn can lead to a large increase in variance if g is not chosen carefully.

2.2 Exponential change of measure

Now we introduce so-called exponential changes of measure. These are very convenient measure transformations that are often used for importance sampling. We roughly stick to the information that can be found in [Glasserman 2003]. Let F be a cumulative distribution function on $\mathbb{R}$.

Definition 2.2.1 The *moment generating function* of F is defined by

$$M(\theta) = \int_{-\infty}^{\infty} e^{\theta x} dF(x).$$

Definition 2.2.2 The *cumulant generating function* of F is defined by

$$\psi(\theta) = \log \int_{-\infty}^{\infty} e^{\theta x} dF(x),$$

the logarithm of the moment generating function of F.

Set $\Theta = \{\theta : \psi(\theta) < \infty\}$ and suppose that Θ is nonempty. For each $\theta \in \Theta$, let us define

$$F_\theta(x) = \int_{-\infty}^{x} e^{\theta u - \psi(\theta)} dF(u).$$

Each F_θ is a probability distribution, and $\{F_\theta, \theta \in \Theta\}$ form an *exponential family* of distributions. A transformation from F to F_θ is called exponential change of measure, exponential tilting or exponential twisting. The density of F_θ is given by

$$f_\theta(x) = e^{\theta x - \psi(\theta)} f(x),$$

where f is supposed to be the density of F.

Let X have initial distribution function $F = F_0$. If we apply an exponential change of measure such that X has the new distribution F_θ, the corresponding likelihood ratio is given by

$$\frac{dF_0(X)}{dF_\theta(X)} = \exp(-\theta X + \psi(\theta)).$$

Note that the cumulant generating function ψ contains important information about the distributions F_θ. For example, $\psi'(\theta)$ is the mean of F_θ and $\psi''(\theta)$ is the variance of F_θ.

Exponential twisting can be applied to every distribution for which a cumulant generating function is defined. Let us for example consider the normal distribution.

Example 2.2.3 The cumulant generating function of the standard normal distribution is given by $\psi(\theta) = \theta^2/2$ and since

$$e^{\theta x - \psi(\theta)} \cdot \phi_{(0,1)} = \phi_{(\theta,1)} \ \forall x \in \mathbb{R}, \ \forall \theta \in \mathbb{R},$$

applying an exponential change of measure to $\mathcal{N}(0,1)$ corresponds to a change of mean from 0 to θ. Here $\phi_{(0,1)}$ and $\phi_{(\theta,1)}$ denote the densities of the normal distributions $\mathcal{N}(0,1)$ and $\mathcal{N}(\theta,1)$, respectively.

If the cumulant generating function of a distribution is not defined, we cannot apply exponential twisting. This is the case for heavy tailed distributions, as for example the t distribution.

To complete this section let us have a look at the following special setting, particularly frequent for option pricing applications. Think of X as a discrete path of underlying assets, built from k primitive elements. Consider $S(t_i)$, $i = 0, 1, \ldots, k$, to be such a path and suppose it is Markov. Let the path be defined by a recursion of the form

$$S(t_{i+1}) = G(S(t_i), X_{i+1}),$$

driven by i.i.d. random vectors $X_1, X_2, \ldots, X_k$, drawn from f. Applying a change of measure that changes the common distribution of $X_1, X_2, \ldots, X_k$ to g while preserving their independence leads to a likelihood ratio given by

$$\prod_{i=1}^{k} \frac{f(X_i)}{g(X_i)}.$$

In the case of an exponential change of measure the likelihood ratio is

$$\prod_{i=1}^{k} \frac{dF_0(X_i)}{dF_\theta(X_i)} = \exp\left(-\theta \sum_{i=1}^{k} X_i + k\psi(\theta) \right),$$

where F_0 is the initial and F_θ the new common distribution of $X_1, X_2, \ldots, X_k$. For a general change of measure the likelihood ratio is a function of all the $X_1, X_2, \ldots, X_k$. The special feature of exponential twisting is that the ratio reduces to a function of the sum $\sum_{i=1}^k X_i$. This setting can again be illustrated using the normal distribution.

Example 2.2.4 Suppose that we want to simulate Brownian motion on a grid $0 = t_0 < t_1 < \ldots < t_k$. Let us set

$$W(t_l) = \sum_{i=1}^{l} \sqrt{t_i - t_{i-1}} X_i, \text{ for } l = 1, 2, \ldots, k,$$

where $X_1, X_2, \ldots, X_k$ are i.i.d. random variables initially drawn from $\phi_{(0,1)}$, the univariate standard normal density. Let $\phi_{(\theta,1)}$ denote the density of $\mathcal{N}(\theta, 1)$. If we apply a change of measure such that $X_1, X_2, \ldots, X_k$ are i.i.d., now drawn from $\phi_{(\theta,1)}$, the corresponding likelihood ratio is

$$\prod_{i=1}^{k} \frac{\phi_{(0,1)}(X_i)}{\phi_{(\theta,1)}(X_i)} = \exp\left(-\theta \sum_{i=1}^{k} X_i + \frac{k}{2}\theta^2 \right) = \exp\left(-\theta \sum_{i=1}^{k} X_i + k\psi(\theta) \right).$$

This change of measure adds mean $\theta\sqrt{t_i - t_{i-1}}$ to the Brownian increment over $[t_{i-1}, t_i]$.

3 Estimating risk measures via Monte Carlo

3.1 Preliminary assumptions

In this section we will describe how to use Monte Carlo methods to estimate loss quantiles or, more specifically, Value-at-Risk and expected shortfall of a financial portfolio. We will treat both, the standard Monte Carlo and the importance sampling case.

Let us make the following model assumptions, identical to those made in [Glasserman et al. 1999]. As already mentioned in the introduction, we want to study $L = -(V_t - V_0)$, the loss of a given portfolio during a time period of length t. V_0 and V_t are again considered to be the initial portfolio value and the portfolio value at time t, respectively. We assume that the portfolio value directly depends on m risk factors. Let $S(0) = (S_1(0), S_2(0), \ldots, S_m(0))$ and $S(t) = (S_1(t), S_2(t), \ldots, S_m(t))$ denote the values of these factors at time 0 and t, respectively. $S(0)$ is supposed to be known, whereas $S(t)$ is unknown. We define $\Delta S := [S(t) - S(0)]^\top$ to be the change in risk factors during the time interval $[0, t]$. The distribution of ΔS is the most important component

of the model assumptions. Let f be the joint density of ΔS. Suppose that the loss L can be computed by applying some deterministic function on ΔS. Based on loss simulations, we want to get useful information about F_L, the unknown distribution function of L.

3.2 Standard Monte Carlo

3.2.1 Estimation of loss probabilities

Using f, we generate N simulations $\Delta S_1, \Delta S_2, \ldots, \Delta S_N$ of changes in risk factors and compute the respective losses $L_1, L_2, \ldots, L_N$. The simulated losses are supposed to have common distribution function F_L. As already mentioned in the introduction, the Monte Carlo estimation of $P(L > x)$ for a fixed loss threshold x is then given by

$$\hat{P}(L > x) = \frac{1}{N} \sum_{i=1}^{N} I\{L_i > x\}. \tag{6}$$

Accordingly, the empirical distribution function of L is

$$\hat{F}_L(x) = \frac{1}{N} \sum_{i=1}^{N} I\{L_i \leq x\}.$$

$\hat{F}_L$ can be used to estimate the loss distribution function F_L, defined by $F_L(x) := P(L \leq x)$.

3.2.2 Estimation of quantiles

Since we are interested in VaR and ES, the estimation of loss probabilities is not enough. We need to estimate quantiles of the loss distribution. The empirical quantile function is the quantile function of $\hat{F}_L$ and thus given by

$$\hat{F}_L^{-1}(p) = \min\{x \in \mathbb{R} : \hat{F}_L(x) \geq p\},$$

for $p \in (0, 1)$. But how to compute $\hat{F}_L^{-1}(p)$? A first method to estimate quantiles of the loss distribution L is suggested by [Glasserman et al. 2002]. According to this method $P(L > x_0)$ is computed for an initial value x_0. Then, a search algorithm is used to choose an updated quantile estimate x_1, depending on the value of $P(L > x_0)$. This procedure is repeated until a predefined level of accuracy has been reached, every probability estimation being based on (6).

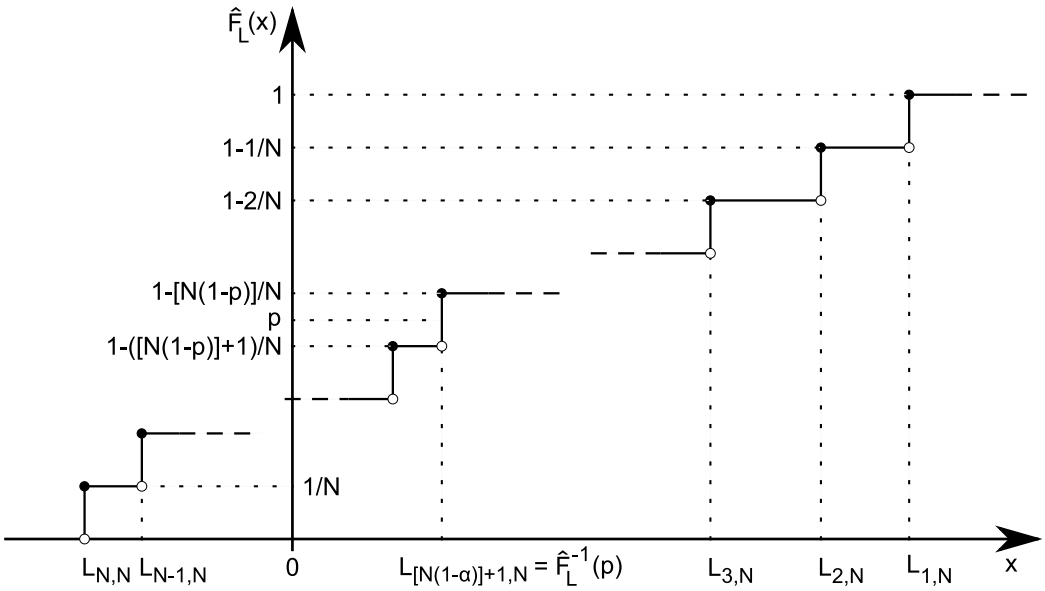

Figure 1: Empirical loss distribution function and its inverse, in the standard Monte Carlo case.

We will focus on a more direct approach, proposed by [Glynn 1996]. If we order the sample $L_1, L_2, \ldots, L_N$ such that $L_{1,N} \geq L_{2,N} \geq \ldots \geq L_{N,N}$, we can write

$$\begin{aligned}\hat{F}_L^{-1}(p) &= \min\left\{x \in \mathbb{R} : \sum_{i=1}^{N} I\{L_{i,N} \leq x\} \geq Np\right\} \\ &= L_{j,N} \text{ for some } j \in \{1, 2, \ldots, N\}.\end{aligned}$$

Since for any $j \in \{1, 2, \ldots, N\}$

$$\sum_{i=1}^{N} I\{L_{i,N} \leq L_{j,N}\} = \sum_{i=j}^{N} I\{L_{i,N} \leq L_{j,N}\} = N - j + 1,$$

we have to find the largest j such that $N - j + 1 \geq Np$. This condition is satisfied by $j = [N(1-p) + 1]$ and therefore, for $p \in (0, 1)$,

$$\hat{F}_L^{-1}(p) = L_{[N(1-p)]+1,N}.$$

$\hat{F}_L^{-1}$ is a piece-wise constant function on $(0, 1)$. More precisely,

$$\hat{F}_L^{-1}(p) = \begin{cases} L_{i,N} & \text{if } p \in (1 - i/N, 1 - (i-1)/N] \text{ for some } i \in \{2, 3, \ldots, N\} \\ L_{1,N} & \text{if } p \in (1 - 1/N, 1). \end{cases}$$

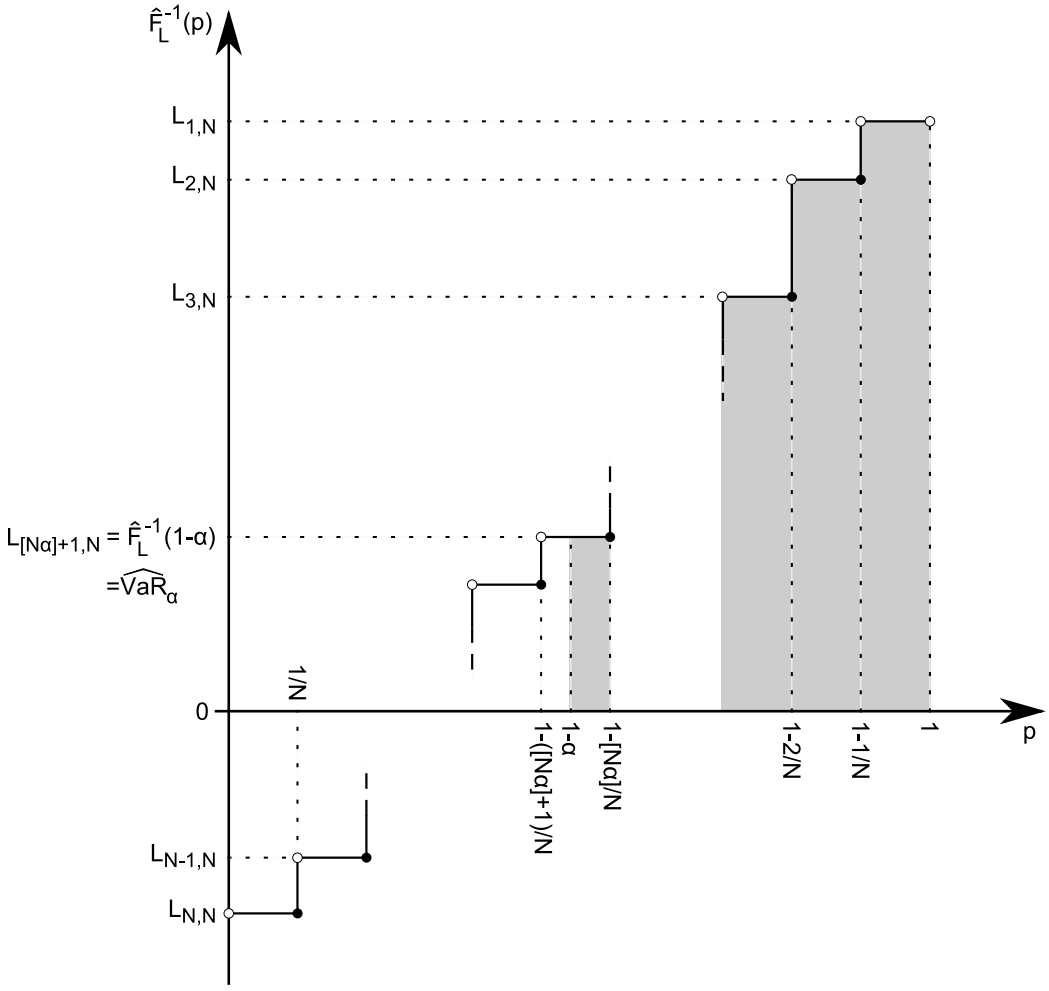

Figure 2: Graph of $\hat{F}_L^{-1}$ and illustration for computation of $\widehat{\mathrm{VaR}}_\alpha$ and $\widehat{\mathrm{ES}}_\alpha$, in the standard Monte Carlo case. The grey area in this figure corresponds to the integral needed to compute $\widehat{\mathrm{ES}}_\alpha$.

The empirical loss distribution function is illustrated in Figure 1. This figure also roughly shows how $\hat{F}_L^{-1}$ is determined. More details on $\hat{F}_L^{-1}$ are shown in Figure 2.

3.2.3 Estimation of Value-at-Risk

According to (1), VaR can be interpreted as a quantile of the loss distribution. Hence we can estimate VaR of our portfolio by

$$\widehat{\mathrm{VaR}}_\alpha = \hat{F}_L^{-1}(1-\alpha) = L_{[N\alpha]+1,N}, \tag{7}$$

for $\alpha \in (0,1)$. A graphical illustration of $\widehat{\mathrm{VaR}}_\alpha$ is given in Figure 2.

3.2.4 Estimation of expected shortfall

By (2), we may estimate expected shortfall of our portfolio by

$$\widehat{\mathrm{ES}}_\alpha = \frac{1}{\alpha} \int_{1-\alpha}^{1} \hat{F}_L^{-1}(p)dp,$$

for $\alpha \in (0,1)$. If we apply the information from Section 3.2.2 to this integral, we can conclude that

$$\widehat{\mathrm{ES}}_\alpha = \frac{1}{\alpha}\left(\sum_{i=1}^{[N\alpha]} \frac{L_{i,N}}{N} + \left(\alpha - \frac{[N\alpha]}{N}\right) L_{[N\alpha]+1,N}\right). \tag{8}$$

The best way to understand this last step is probably to have a look at Figure 2, which illustrates the computation of $\widehat{\mathrm{ES}}_\alpha$.

3.3 Monte Carlo and importance sampling

3.3.1 Estimation of loss probabilities

Let $\Delta S_1, \Delta S_2, \ldots, \Delta S_N$ now be independent simulations drawn from the densities $g_1, g_2, \ldots, g_N$, respectively. The corresponding losses are again called $L_1, L_2, \ldots, L_N$. The new densities are chosen in such a way that more values from the tail of F_L are generated than under the initial density f. That is, we put more "importance" on large losses. Note that, referring to Remark 2.1.1, we use N importance sampling densities instead of only one. This offers the possibility to update the importance sampling density while generating simulations and may, in some cases, lead to an even more efficient estimator.

More information on how to choose the g_i in practice will follow in Section 4 and Section 5.

Let us now estimate $P(L > x) = E[I\{L > x\}]$, for some fixed loss threshold x. If we set $h(\Delta S) = I\{L > x\}$, estimating loss probabilities is equivalent to estimating $E[h(\Delta S)]$. Plugging $h(\Delta S_i) = I\{L_i > x\}$ into (5) leads to

$$\hat{P}(L > x) = \frac{1}{N} \sum_{i=1}^{N} I\{L_i > x\} \, \frac{f(\Delta S_i)}{g_i(\Delta S_i)} =: \sum_{i=1}^{N} w_i \, I\{L_i > x\}. \tag{9}$$

Importance sampling puts special emphasis on the tail of the loss distribution. Therefore, instead of working with the empirical loss distribution function, we will subsequently use the following *empirical importance sampling tail function*

$$\hat{G}_L(x) := \hat{P}(L > x) = \sum_{i=1}^{N} w_i \, I\{L_i > x\}.$$

Note that in general $\sum_{i=1}^{N} w_i \neq 1$. That is, normalization, in the sense of $\hat{P}(L > x) = 1 - \hat{P}(L \leq x)$, is not satisfied here and accordingly,

$$\hat{G}_L(x) \neq 1 - \sum_{i=1}^{N} w_i \, I\{L_i \leq x\},$$

in general. We introduce the somewhat special name of empirical "importance sampling" tail function to distinguish $\hat{G}_L$ from the usual empirical tail function.

3.3.2 Estimation of "quantiles"

As in Section 3.2.2, let us again order the sample $L_1, L_2, \ldots, L_N$ such that $L_{1,N} \geq L_{2,N} \geq \ldots \geq L_{N,N}$. The order of $\Delta S_1, \Delta S_2, \ldots, \Delta S_N$ is changed in such a way that $L_{i,N}$ is the loss corresponding to $\Delta S_{i,N}$. The density of $\Delta S_{i,N}$ is supposed to be $g_{i,N}$ and finally $w_{i,N} = \frac{1}{N} \cdot \frac{f(\Delta S_{i,N})}{g_{i,N}(\Delta S_{i,N})}$. Now, $\hat{G}_L(x)$ can be reformulated as

$$\hat{G}_L(x) = \sum_{i=1}^{N} w_{i,N} \, I\{L_{i,N} > x\}.$$

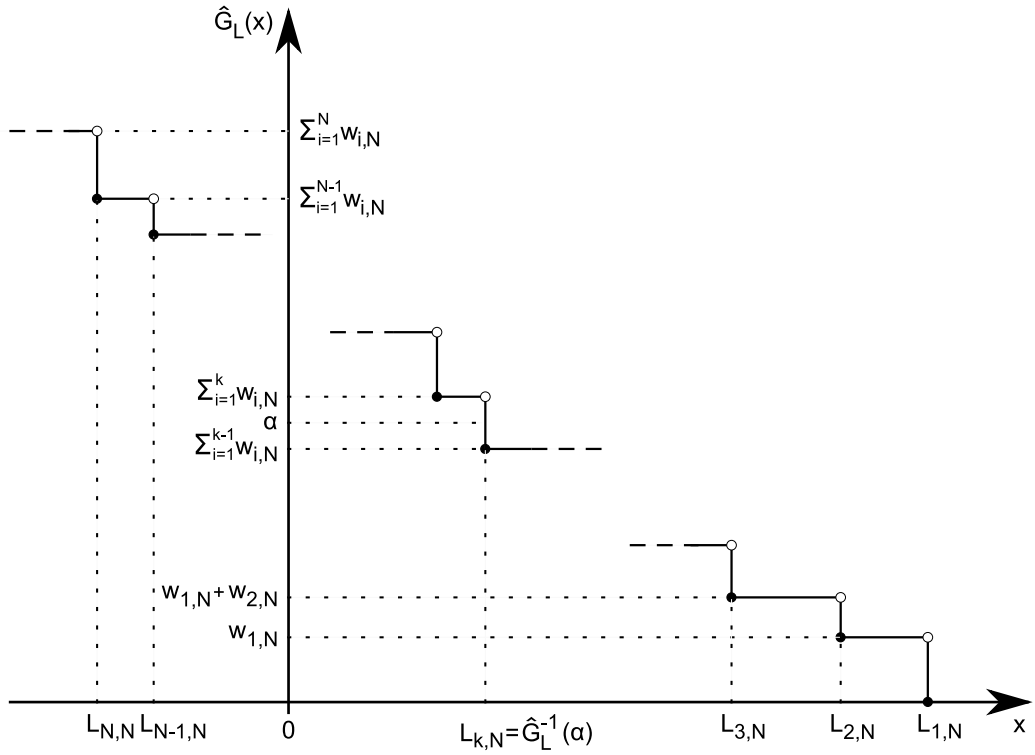

Figure 3: Empirical importance sampling tail function $\hat{G}_L$ and its inverse. Note that in this figure $k = \min\left\{ j : \sum_{i=1}^{j} w_{i,N} \geq \alpha \right\}$.

Since $\hat{G}_L$ is a decreasing positive function that goes to zero, its "inverse", for $\alpha \in (0,1)$, can be written as

$$\begin{aligned}\hat{G}_L^{-1}(\alpha) &= \min\left\{x \in \mathbb{R} : \sum_{i=1}^{N} w_{i,N}\ I\{L_{i,N} > x\} \leq \alpha\right\} \\ &= L_{j,N} \text{ for some } j \in \{1,2,\ldots,N\}.\end{aligned}$$

For any $j \in \{1,2,\ldots,N\}$

$$\sum_{i=1}^{N} w_{i,N}\ I\{L_{i,N} > L_{j,N}\} = \sum_{i=1}^{j-1} w_{i,N}\ I\{L_{i,N} > L_{j,N}\} = \sum_{i=1}^{j-1} w_{i,N}.$$

That is, we have to find the largest j such that $\sum_{i=1}^{j-1} w_{i,N} \leq \alpha$ or, equivalently, the smallest j such that $\sum_{i=1}^{j} w_{i,N} \geq \alpha$. Hence,

$$\hat{G}_L^{-1}(\alpha) = L_{k,N} \text{ where } k = \min\left\{j : \sum_{i=1}^{j} w_{i,N} \geq \alpha\right\},\ \alpha \in (0,1). \tag{10}$$

$\hat{G}_L^{-1}$ is a piece-wise constant function:

$$\hat{G}_L^{-1}(\alpha) = \begin{cases} L_{i,N} & \text{if } \alpha \in \left[\sum_{j=1}^{i-1} w_{j,N}, \sum_{j=1}^{i} w_{j,N}\right) \text{ for some } i \in \{2,3,\ldots,N\} \\ L_{1,N} & \text{if } \alpha \in (0, w_{1,N}). \end{cases}$$

The so-called empirical importance sampling tail function $\hat{G}_L$ is illustrated in Figure 3. This figure also gives a rough illustration on how $\hat{G}_L^{-1}$ is determined. The graph of $\hat{G}_L^{-1}$ is shown in Figure 4.

3.3.3 Estimation of Value-at-Risk

Using (10), VaR can now be estimated by

$$\widehat{\mathrm{VaR}}_\alpha = \hat{G}_L^{-1}(\alpha) = L_{k,N} \text{ where } k = \min\left\{j : \sum_{i=1}^{j} w_{i,N} \geq \alpha\right\}, \tag{11}$$

for $\alpha \in (0,1)$. Figure 4 shows graphically how $\widehat{\mathrm{VaR}}_\alpha$ is determined.

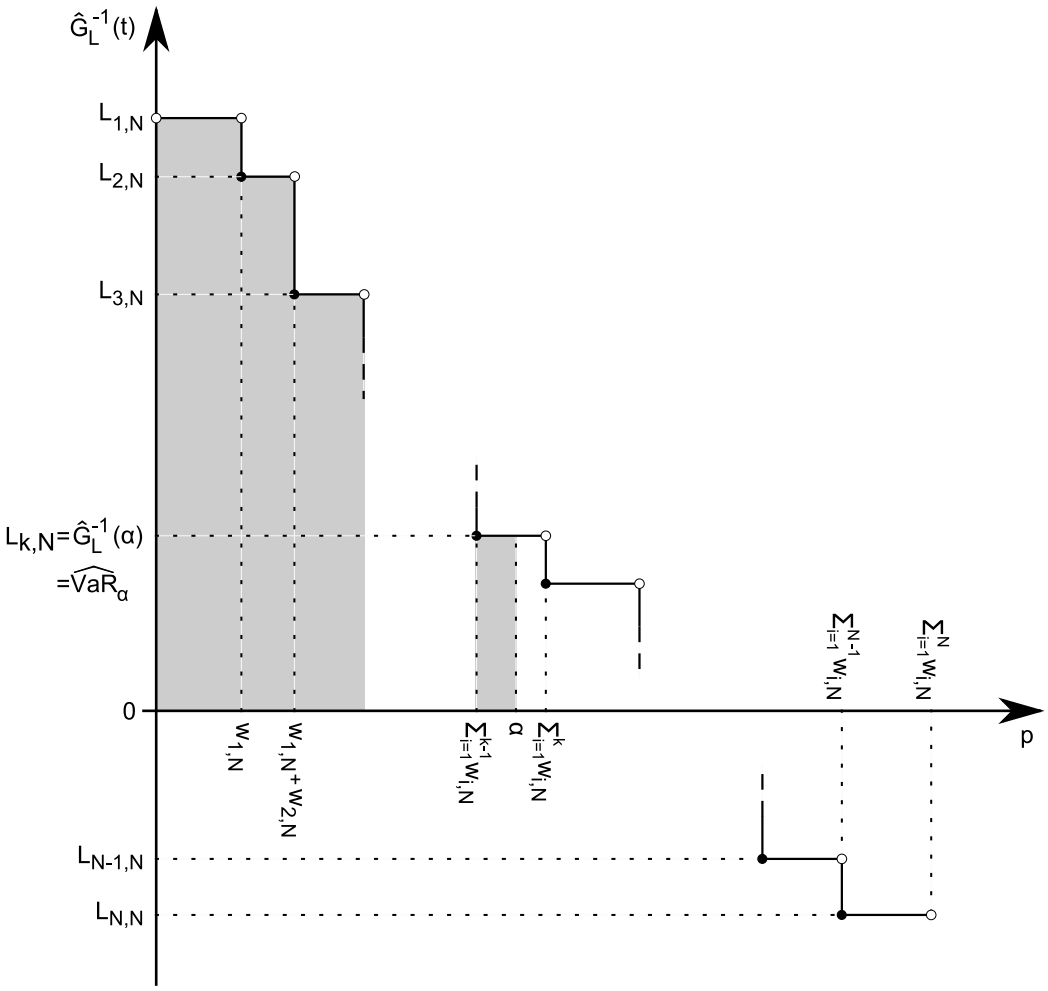

Figure 4: Graph of $\hat{G}_L^{-1}$ and illustration of the computation of $\widehat{\mathrm{VaR}}_\alpha$ and $\widehat{\mathrm{ES}}_\alpha$, with importance sampling. The grey area in this figure corresponds to the integral needed to compute $\widehat{\mathrm{ES}}_\alpha$. Here $k = \min\left\{ j : \sum_{i=1}^{j} w_{i,N} \geq \alpha \right\}$, as explained in Section 3.3.4.

3.3.4 Estimation of expected shortfall

With the information on $\hat{G}_L^{-1}$ from Section 3.3.2, we can now estimate ES at level $\alpha \in (0,1)$:

$$\begin{aligned}\widehat{\mathrm{ES}}_\alpha &= \frac{1}{\alpha}\int_0^\alpha \hat{G}_L^{-1}(t)dt \\ &= \frac{1}{\alpha}\left(\sum_{i=1}^{k-1} w_{i,N}\ L_{i,N}\ + \left(\alpha - \sum_{i=1}^{k-1} w_{i,N}\right) L_{k,N}\right), \end{aligned} \tag{12}$$

where $k = \min\left\{j : \sum_{i=1}^{j} w_{i,N} \geq \alpha\right\}$. Figure 4 gives a good illustration of the integration needed for the computation of $\widehat{\mathrm{ES}}_\alpha$.

4 Loss probability estimations by importance sampling in the light tailed case

4.1 The delta and delta-gamma approximations

Let us stick to the assumptions made in Section 3.1 and assume that the ΔS has a light tailed distribution, more specifically, a multivariate normal distribution with mean vector zero and covariance matrix Σ. That is, we assume that $\Delta S \sim \mathcal{N}(0,\Sigma)$. Simulating ΔS can be reduced to simulating multivariate standard normals $Z \sim \mathcal{N}(0,I)$, by setting

$$\Delta S = CZ, \text{ with } CC^\top = \Sigma.$$

Suppose that we can get L from Z using

$$L = f^L(Z), \tag{13}$$

where f^L is a deterministic function.

To be able to use importance sampling for loss probability estimations, an importance sampling density has to be chosen. For the selection of an efficient density, information about the loss distribution F_L is required. Since F_L is unknown, a logical approach is to exploit knowledge of the distribution of an approximation to the loss. In [Glasserman et al. 1999] the choice of efficient importance sampling densities is based on first and second order Taylor expansions, also called delta and delta-gamma approximations. We will make a review of this approach, based on information from [Glasserman et al. 1999] and [Glasserman 2003]. Let us first describe both approximations in detail.

The delta-gamma approximation to the loss L is given by

$$L = -\Delta V \approx -\left(\frac{\partial V}{\partial t}t + \delta^\top \Delta S + \frac{1}{2}\Delta S^\top \Gamma \Delta S\right), \tag{14}$$

where $\delta_i = \frac{\partial V}{\partial S_i(t)}$ and $\Gamma_{ij} = \frac{\partial^2 V}{\partial S_i(t)\partial S_j(t)}$. The partial derivatives are supposed to be evaluated at $S(0)$. In terms of Z the delta-gamma approximation can be written as

$$L \approx -\frac{\partial V}{\partial t}t - \left(C^\top \delta\right)^\top Z - \frac{1}{2}Z^\top \left(C^\top \Gamma C\right) Z. \tag{15}$$

Working with the delta-gamma approximation is much easier if we choose C to diagonalize $\frac{1}{2}Z^\top \left(C^\top \Gamma C\right) Z$. We will now show that this choice is possible. Let $\tilde{C}$ be a square matrix such that $\tilde{C}\tilde{C}^\top = \Sigma$. More specifically, set $\tilde{C}$ to be the lower triangular matrix found by Cholesky factorization of the symmetric covariance matrix Σ. Then, $-\frac{1}{2}\tilde{C}^\top \Gamma \tilde{C}$ is symmetric and can be written as

$$-\frac{1}{2}\tilde{C}^\top \Gamma \tilde{C} = U \Lambda U^\top,$$

where

$$\Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{pmatrix}$$

is a diagonal matrix and U is an orthogonal matrix whose columns are eigenvectors of $-\frac{1}{2}\tilde{C}^\top \Gamma \tilde{C}$. The corresponding eigenvalues are $\lambda_1, \lambda_2, \ldots, \lambda_m$, the diagonal elements of Λ. Let us now show that $C := \tilde{C}U$ satisfies the required conditions for diagonalizing the quadratic term in (15). The first condition,

$$CC^\top = \tilde{C}UU^\top \tilde{C}^\top \overset{(U \text{ orthogonal})}{=} \Sigma,$$

is verified and since

$$-\frac{1}{2}C^\top \Gamma C = -\frac{1}{2}U^\top \left(\tilde{C}^\top \Gamma \tilde{C}\right) U = U^\top \left(U \Lambda U^\top\right) U \overset{(U \text{ orthogonal})}{=} \Lambda,$$

the second is as well. If we set $a = -\frac{\partial V}{\partial t}t$ and $b = -C^\top \delta$ we can reformulate (15) as

$$L \approx Q := a + b^\top Z + Z^\top \Lambda Z = a + \sum_{j=1}^{m} \left(b_j Z_j + \lambda_j Z_j^2\right). \tag{16}$$

Since $-\frac{1}{2}\tilde{C}^\top\Gamma\tilde{C} = -\frac{1}{2}\Gamma\tilde{C}\tilde{C}^\top = -\frac{1}{2}\Gamma\Sigma$, $\lambda_1, \lambda_2, \ldots, \lambda_m$ are considered to be the eigenvalues of $-\frac{1}{2}\Gamma\Sigma$.

Similarly to the delta-gamma approximation in (14), the delta approximation is given by

$$L = -\Delta V \approx -\left(\frac{\partial V}{\partial t}t + \delta^\top \Delta S\right).$$

Using the notation introduced above, we get the following more convenient representation

$$L \approx Y := a + b^\top Z = a + \sum_{j=1}^{m} b_j Z_j \tag{17}$$

for the first order Taylor expansion of the loss L. As a linear combination of m independent standard normal random variables, $\sum_{j=1}^{m} b_j Z_j$ is again normally distributed. We can conclude that

$$P(Y \geq x) = 1 - \Phi_{(a, \sum_{j=1}^{m} b_j^2)}(x), \tag{18}$$

where $\Phi_{(0, \sum_{j=1}^{m} b_j^2)}$ is the cumulative distribution function of the normal distribution with mean a and variance $\sum_{j=1}^{m} b_j^2$.

Let us now come back to the delta-gamma approximation. For the distribution of Q there is no closed form, but we will now introduce a method to compute it numerically. By assuming that $\lambda_i \neq 0$ for $i = 1, 2, \ldots, m$ and completing the square, we can write Q as

$$Q = a + \sum_{j=1}^{m} \left(\lambda_j \left(Z_j + \frac{b_j}{2\lambda_j}\right)^2 - \frac{b_j^2}{4\lambda_j}\right). \tag{19}$$

Let $k \leq m$ denote the number of distinct eigenvalues of $-\frac{1}{2}\Gamma\Sigma$ and consider $h_1, h_2, \ldots, h_k$ to be their respective algebraic multiplicities. We reorder the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ such that for

$$\lambda_{1,1}, \lambda_{1,2}, \ldots, \lambda_{1,h_1}, \lambda_{2,1}, \lambda_{2,2}, \ldots, \lambda_{2,h_2}, \ldots, \lambda_{k,1}, \lambda_{k,2}, \ldots, \lambda_{k,h_k}$$

the following two conditions are satisfied for each $j \in \{1, 2, \ldots, k\}$:

- $\lambda_{j,i} = \lambda_{j,l}\ \forall i \in \{1, 2, \ldots, h_j\} \forall l \in \{1, 2, \ldots, h_j\}$
- $\lambda_{j,i} \neq \lambda_{r,l}\ \forall r \in \{\{1, 2, \ldots, k\} \setminus j\}\, \forall i \in \{1, 2, \ldots, h_j\} \forall l \in \{1, 2, \ldots, h_r\}$.

This denomination scheme, based on a double index, ensures that we can still access to all of the m eigenvalues. Let us denominate the elements of b and Z accordingly and apply the new notation to (19):

$$Q = a + \sum_{j=1}^{k} \sum_{i=1}^{h_j} \left(\lambda_{j,i} \left(Z_{j,i} + \frac{b_{j,i}}{2\lambda_{j,i}} \right)^2 - \frac{b_{j,i}^2}{4\lambda_{j,i}} \right).$$

For each fixed $j \in \{1, 2, \ldots, k\}$,

$$\sum_{i=1}^{h_j} \lambda_{j,i} \left(Z_{j,i} + \frac{b_{j,i}}{2\lambda_{j,i}} \right)^2 = \lambda_{j,1} \sum_{i=1}^{h_j} \left(Z_{j,i} + \frac{b_{j,i}}{2\lambda_{j,i}} \right)^2 = \lambda_{j,1} \; \chi^2_{h_j, d_j^2},$$

where $\chi^2_{h_j,d_j^2}$ is a non-central χ^2-distribution with h_j degrees of freedom and non-centrality parameter

$$d_j^2 = \sum_{i=1}^{h_j} \left(\frac{b_{j,i}}{2\lambda_{j,i}} \right)^2.$$

This allows us to write

$$Q = a - \sum_{j=1}^{k} \sum_{i=1}^{h_j} \frac{b_{j,i}^2}{4\lambda_{j,i}} + \sum_{j=1}^{k} \lambda_{j,1} \; \chi^2_{h_j, d_j^2}.$$

Note that $\lambda_{1,1}, \lambda_{2,1}, \ldots, \lambda_{k,1}$ are the k distinct eigenvalues of $-\frac{1}{2}\Gamma\Sigma$. Now we have reduced the distribution of Q to a linear combination of non-central χ^2-distributions. There is a procedure, suggested by [Imhof 1961], that allows to numerically compute the distribution of

$$R := \sum_{j=1}^{k} \lambda_{j,1} \; \chi^2_{h_j, d_j^2},$$

by inversion of the characteristic function. The characteristic function of R is given by

$$\phi(t) = \prod_{j=1}^{k} (1 - 2i\lambda_{j,1}t)^{-\frac{1}{2}h_j} \exp\left(i \sum_{j=1}^{k} \frac{d_j^2 \lambda_{j,1} t}{1 - 2i\lambda_{j,1}t} \right),$$

where $i = \sqrt{-1}$. According to [Gurland 1948] and [Gil-Palaez 1951], an inversion formula of $\phi(t)$ is given by

$$P(R \leq x) = \frac{1}{2} - \frac{1}{\pi} \int_0^\infty t^{-1} \mathrm{Im}\left(e^{-itx} \phi(t) \right) dt,$$

where $\mathrm{Im}(z)$ denotes the imaginary part of z. This can be written as

$$P(R \leq x) = \frac{1}{2} - \frac{1}{\pi} \int_0^\infty \frac{\sin\left(\tau(u)\right)}{u\rho(u)} du, \tag{20}$$

where

$$\tau(u) = \frac{1}{2} \sum_{j=1}^{k} \left(h_j \tan^{-1}(\lambda_{j,1} u) + d_j^2 \lambda_{j,1} u \left(1 + \lambda_{j,1}^2 u^2\right)^{-1} \right) - \frac{1}{2} x u,$$

$$\rho(u) = \prod_{j=1}^{k} \left(1 + \lambda_{j,1}^2 u^2\right)^{\frac{1}{4} h_j} \exp\left(\frac{\frac{1}{2} \sum_{j=1}^{k} (d_j \lambda_{j,1} u)^2}{\left(1 + \lambda_{j,1}^2 u^2\right)} \right).$$

The inversion integral in (20) allows to numerically compute $P(Q > x)$ or even quantiles of Q. Additionally, using (18), we could easily estimate loss probabilities and quantiles of F_L, based on the delta approximation. But these approximations are normally not accurate enough to provide precise risk measure estimations. Nevertheless we can exploit information about Y and Q to choose an efficient importance sampling density.

4.2 Importance sampling density based on the delta approximation

According to [Bucklew 1990] the choice of an effective importance sampling density for the estimation of $P(L > x)$, for a large threshold x, can be based on a large deviations analysis. More precisely, the probability of a rare event is approximately equal to the probability of "the most likely path" to this rare event. Let us assume that loss changes are linear in ΔS. That is, $L \approx Y$ in (17) is supposed to be exact. We consider $E = \left\{ z \in \mathbb{R}^m : b^\top z > x - a \right\}$ to be the set of rare events that should be given more weight under the importance sampling distribution than under the original multivariate standard normal distribution. Let $\mu = (\mu_1, \mu_2, \ldots, \mu_m)^\top \in \mathbb{R}^m$ be the point that maximizes the probability of a rare event under the original distribution. The most probable value of a random variable corresponds to the value that maximizes its density function. In our case, μ is the solution of the optimization problem

$$\max -\tfrac{1}{2} \sum_{i=1}^{m} z_i^2 \quad \text{such that} \quad \sum_{i=1}^{m} b_i z_i \geq x - a,$$

because the original distribution is a multivariate standard normal distribution. The solution to this optimization problem is

$$\mu = (x - a) \frac{b}{b^\top b}. \tag{21}$$

Since an effective importance sampling density is supposed to make a rare event more likely, it should in our situation put more weight on μ, being the most likely path to E. According to [Chen et al. 1993] an exponentially efficient importance sampling density is obtained by changing the mean of Z from 0 to μ. A simulation scheme is considered to be *exponentially efficient* if the number of simulations required to obtain a specified precision grows less than exponentially fast. Instead of $Z \sim \mathcal{N}(0, I)$, we will therefore use $Z \sim \mathcal{N}(\mu, I)$ for estimating $P(L > x)$. The corresponding likelihood ratio is given by

$$\frac{f(Z)}{g(Z)} = \exp\left(\frac{1}{2}\mu^\top \mu - \mu^\top Z\right), \tag{22}$$

where f and g are the densities of $\mathcal{N}(0, I)$ and $\mathcal{N}(\mu, I)$, respectively. The per sample second moment of the resulting importance sampling estimator is

$$\tilde{\mathrm{E}}\left[I\{L > x\}\left(\frac{f(Z)}{g(Z)}\right)^2\right] = \mathrm{E}\left[I\{L > x\}\frac{f(Z)}{g(Z)}\right].$$

A sufficient condition for variance reduction is therefore given by

$$\frac{f(Z)}{g(Z)} \leq 1 \; \forall Z \in E.$$

How can this condition be interpreted?

$$\begin{aligned}\frac{f(Z)}{g(Z)} \leq 1 &\Leftrightarrow \frac{1}{2}\mu^\top \mu - \mu^\top Z \leq 0 \\ &\Leftrightarrow \mu^\top \mu - 2\mu^\top Z + Z^\top Z \leq Z^\top Z \\ &\Leftrightarrow \|\mu - Z\| \leq \|Z\|.\end{aligned}$$

That is, if every point $Z \in E$ is closer to μ than to the origin, variance is reduced. Note that if $\frac{f(Z)}{g(Z)} \leq f \; \forall Z \in E$, then the second moment of the importance sampling estimator is reduced by at least a factor of f.

4.3 Importance sampling density based on the delta-gamma approximation

Let us now apply what we learned in Section 2.2 and consider importance sampling based on exponential changes of measure. Since we have no knowledge about L itself, we will exponentially twist the delta-gamma approximation Q. Let us first determine the cumulant generating function of Q. By

making use of (19), we can write the moment generating function of Q as

$$\begin{aligned}\mathrm{E}\left[\exp(\theta Q)\right] &= \exp(\theta a)\prod_{j=1}^{m}\mathrm{E}\left[\exp\left(\theta\left(\lambda_j\left(Z_j+\frac{b_j}{2\lambda_j}\right)^2-\frac{b_j^2}{4\lambda_j}\right)\right)\right]\\ &= \exp\left(\theta a\right)\prod_{j=1}^{m}\exp\left(-\theta\frac{b_j^2}{4\lambda_j}\right)\mathrm{E}\left[\exp\left(\theta\lambda_j\left(Z_j+\frac{b_j}{2\lambda_j}\right)^2\right)\right].\end{aligned}$$

Note that $(Z_j + b_j/(2\lambda_j))^2$ is a non-central χ^2-distribution with one degree of freedom and non-centrality parameter $(b_j/(2\lambda_j))^2$. From equation (29.6) of [Johnson et al. 1995],

$$\mathrm{E}\left[\exp\left(\theta\left(Z_j+c\right)^2\right)\right] = \left(1-2\theta\right)^{-\frac{1}{2}}\exp\left(\frac{\theta c^2}{1-2\theta}\right),$$

for $\theta < 1/2$, follows

$$\mathrm{E}\left[\exp\left(\theta\lambda_j\left(Z_j+\frac{b_j}{2\lambda_j}\right)^2\right)\right] = \left(1-2\theta\lambda_j\right)^{-\frac{1}{2}}\exp\left(\frac{\theta\frac{b_j^2}{4\lambda_j}}{1-2\theta\lambda_j}\right),$$

for θ such that $\theta\lambda_j < 1/2$. Hence, the cumulant generating function of Q is given by

$$\begin{aligned}\psi(\theta) &= \log\left(\mathrm{E}\left[\exp(\theta Q)\right]\right)\\ &= \theta a+\sum_{j=1}^{m}\left(-\theta\frac{b_j^2}{4\lambda_j}-\frac{\log\left(1-2\theta\lambda_j\right)}{2}+\frac{\theta\frac{b_j^2}{4\lambda_j}}{1-2\theta\lambda_j}\right)\\ &= \theta a+\frac{1}{2}\sum_{j=1}^{m}\left(\frac{\theta^2 b_j^2}{1-2\theta\lambda_j}-\log\left(1-2\theta\lambda_j\right)\right),\end{aligned} \tag{23}$$

for $\theta \in \Theta = \{\theta : \psi(\theta) < \infty\} = \{\theta : \max_j \theta\lambda_j < 1/2\}$.

Now we can define a new probability measure P_θ, for $\theta \in \Theta$, through the likelihood ratio

$$\frac{dP_0}{dP_\theta} = \exp(-\theta Q + \psi(\theta)). \tag{24}$$

The resulting importance sampling estimator for $\alpha = P(L > x)$ is

$$\hat{\alpha}_\theta = \frac{1}{N}\sum_{i=1}^{N} I\{L_i > x\}\,\exp(-\theta Q_i + \psi(\theta)), \tag{25}$$

where L_i and Q_i are being computed from the i-th simulation of Z, using (13) and (16), respectively. Further details on the distribution of Z under the new measure will follow below. Note that the new probability measure P_θ, defined by (24), increases the probability of the events $\{L : Q > x\}$, if $\theta > 0$. This seems intuitively appealing, assuming that Q is an accurate loss approximation. Moreover, if $\theta > 0$ and $L \approx Q$ is exact, the second moment of $\hat{\alpha}_\theta$ is exponentially decreasing in x. Indeed,

$$\mathrm{E}_\theta\left[\left(I\{L > x\}e^{-\theta Q+\psi(\theta)}\right)^2\right] = \mathrm{E}\left[I\{L > x\}e^{-\theta Q+\psi(\theta)}\right] \leq e^{-\theta x+\psi(\theta)}, \qquad (26)$$

which, for $\theta > 0$, decreases exponentially in x.

In practice it is important to be able to sample from the importance sampling distribution defined by (24). More precisely, to make use of the estimator in (25) we need to generate N simulations of Z from its distribution under the new measure. We will now show that, under P_θ, $Z \sim \mathcal{N}(\mu(\theta), \Sigma(\theta))$, where

$$\mu_j(\theta) = \frac{\theta b_j}{1 - 2\lambda_j\theta} \qquad (27)$$

and $\Sigma(\theta)$ is a diagonal matrix with diagonal elements

$$\sigma_j^2(\theta) = \frac{1}{1 - 2\lambda_j\theta}, \qquad (28)$$

for all $\theta \in \Theta = \{\theta : \max_j \theta\lambda_j < 1/2\}$. The likelihood ratio corresponding to the change of measure from $\mathcal{N}(0, I)$ to $\mathcal{N}(\mu(\theta), \Sigma(\theta))$ can easily be reduced from

$$\frac{|\Sigma(\theta)|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(Z - \mu(\theta))^\top \Sigma(\theta)^{-1}(Z - \mu(\theta))\right)}{\exp\left(-\frac{1}{2}Z^\top Z\right)}$$

to $\exp(\theta Q - \psi(\theta))$, by plugging in (27) and (28) and further making use of equations (16) and (23). Since this is exactly the likelihood ratio used to define P_θ in (24), we can conclude that $\mathcal{N}(\mu(\theta), \Sigma(\theta))$ is the right choice for the distribution of Z under the new measure. Furthermore, considering the delta-gamma approximation in (16), a properly chosen importance sampling distribution should give positive (negative) mean to those Z_j for which $b_j > 0$ ($b_j < 0$) and increase the variance of Z_j if $\lambda_j > 0$. For positive $\theta \in \Theta$, these qualitative conditions are satisfied by (27) and (28). Note that in practice the likelihood ratio $\exp(\theta Q - \psi(\theta))$ is very convenient, because instead of depending on the whole random vector Z it only depends on the value of Q.

A very important component in (25) is an optimal, or at least an efficient, twisting parameter θ. In (26) we introduced an upper bound for the second moment of the importance sampling estimator, under the assumption that

the loss is quadratic. For fixed x, let us choose $\theta = \theta_x$ to minimize this upper bound. Since $\psi(\theta)$ is the cumulant generating function of Q, it is convex. Hence, $-\theta x + \psi(\theta)$ is minimized if $-x + \psi'(\theta) = 0$. That is, θ_x is the root of the nonlinear equation

$$\psi'(\theta_x) = x, \tag{29}$$

which may be computed numerically. According to what has been mentioned in Section 2.2,

$$\psi'(\theta) = \mathrm{E}_\theta[Q] \ \forall \theta \in \Theta.$$

Since $\psi'(\theta_x) = x$, we can say that, under the importance sampling distribution P_{θ_x}, the expected value of Q equals x. This in turn makes high losses very probable under P_{θ_x}.

Let us now present some asymptotic optimality results for importance sampling estimators using exponential twisting, that have been stated and proved in [Glasserman et al. 2000]. We first consider the situation where

$$\lambda_{\max} := \max_{1 \le j \le m} \lambda_j > 0.$$

In this case Q can grow unboundedly and we will therefore analyze asymptotic behavior as $x \to \infty$. Let $m_2(x, \theta_x)$ be the second moment of an estimator using exponential twisting with twisting parameter θ_x. Here, an estimator for $P(L > x)$ is said to be *asymptotically optimal* if there is a constant $c > 0$ such that $P(L > x) = \exp(-cx + o(x))$ and $m_2(x, \theta_x) = \exp(-2cx + o(x))$. This condition is equivalent to m_2 decreasing at twice the exponential rate of the loss probability. Since the square function is convex, Jensen's inequality proves that this is the fastest possible rate for any unbiased estimator. Optimal rates of decrease imply that $m_2(x, \theta_x) \approx P(L > x)^2$, whereas for the standard loss estimator $m_2(x, 0) = \mathrm{E}\left[I\{L > x\}\right] = P(L > x)$. If $\lambda_{\max} > 0$ and $L = Q$, then, according to Theorem 1 and Theorem 2 of [Glasserman et al. 2000],

$$\lim_{x \to \infty} \frac{\log(P(L > x))}{x} = -\frac{1}{2\lambda_{\max}}$$
$$\lim_{x \to \infty} \frac{\log(m_2(x, \theta_x))}{x} = -\frac{1}{\lambda_{\max}},$$

where the twisting parameter θ_x has been chosen according to (29). Hence, $P(L > x) = \exp(-x/(2\lambda_{\max}) + o(x))$ and $m_2(x, \theta_x) = \exp(-x/\lambda_{\max} + o(x))$. This in turn means that, under the above assumptions, exponential twisting with twisting parameter θ_x defined by (29) is asymptotically optimal.

Now we consider the case where

$$\lambda_{\max} < 0.$$

That is, we assume that all the eigenvalues of $-\frac{1}{2}\Gamma\Sigma$ are negative. From (19) follows that in this case Q is bounded by the constant $d := a - \sum_{j=1}^{m} \frac{b_j^2}{4\lambda_j}$. Let $x_\epsilon = d - \epsilon$, then

$$P(L > x_\epsilon) \approx P(Q > x_\epsilon) \to 0 \text{ as } \epsilon \to 0.$$

So, in this case we study asymptotic behavior as $\epsilon \to 0$ or, equivalently, as $x_\epsilon \to d$. If $L = Q$ and $\lambda_{\max} < 0$, then, according to Theorem 3 of [Glasserman et al. 2000], there exist constants $k_1 > 0$ and $k_2 > 0$ such that

$$k_1 \leq \liminf_{\epsilon \to 0} \frac{P(L > x_\epsilon)}{\epsilon^{m/2}} \leq \limsup_{\epsilon \to 0} \frac{P(L > x_\epsilon)}{\epsilon^{m/2}} \leq k_2. \tag{30}$$

Furthermore, for $\theta_\epsilon > 0$ solving $\psi'(\theta_\epsilon) = d - \epsilon$, there exists a constant $k_3 > 0$ such that

$$\limsup_{\epsilon \to 0} \frac{m_2(x_\epsilon, \theta_\epsilon)}{\epsilon^m} \leq k_3. \tag{31}$$

That is, the estimated probability is of order $\epsilon^{m/2}$ and the second moment is of order ϵ^m, which is the best possible exponent. This implies an even stronger form of asymptotic optimality than before. Let the *relative error* (RE) of an estimation be the ratio of the standard deviation and the mean of the estimate. In the case where the loss probability $p := P(L > x)$ is being estimated by $\hat{p}$, RE can be written as

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

The bounds in (30) and (31) imply the so-called *bounded relative error* property, that is, RE remains bounded as $\epsilon \to 0$.

5 From loss probabilities to quantiles

In Section 4 we introduced two different methods of choosing an efficient importance sampling distribution when estimating loss probabilities $P(L > x)$. For the estimation of VaR and ES we need to approximate extreme quantiles of the loss distribution. That is, we are looking for x such that $P(L > x) = p$, for fixed small p. The selection of an importance sampling density for estimating $P(L > x)$ has been based on the threshold value x. For quantile estimation this is not possible, because the exact threshold is a priori unknown. But according to [Glynn 1996], the variance reduction of an importance sampling estimator for $P(L > x)$ carries over to quantile estimation,

Algo1Y$(\alpha, x_{\text{start}}, N)$
$x := x_{\text{start}}$
$\mu := (x-a)\frac{b}{b^\top b}$, by (21)
$A :=$ $N \times 2$ matrix consisting of all zeros
for $i = 1 : N$
    generate $Z \sim \mathcal{N}(\mu, I)$
    $L := f^L(Z)$, loss as in (13)
    $r := \exp\left(\frac{1}{2}\mu^\top\mu - \mu^\top Z\right)$, likelihood ratio as in (22)
    $A(i,:) := (L, r)$
end
sort rows of A in descending order with respect to the first column
$k := \min\left\{j \in \{1, 2, \ldots, N\} : \frac{1}{N}\sum_{l=1}^{j} A(l,2) \geq \alpha\right\}$
$\widehat{\text{VaR}}_\alpha := A(k,1)$, by (11)
$\widehat{\text{ES}}_\alpha := \frac{1}{\alpha}\left(\frac{1}{N}\sum_{l=1}^{k-1} A(l,1)A(l,2) + \left(\alpha - \frac{1}{N}\sum_{l=1}^{k-1} A(l,2)\right)A(k,1)\right)$, by (12)

Figure 5: Algorithm for the estimation of VaR_α and ES_α using delta IS, based on a sample of N simulations. The algorithm does not update the importance sampling density throughout the simulation process. x_{start} is supposed to be a rough guess for VaR_α.

if x is in a neighborhood of the quantile of interest. And by Theorem 5 of [Glasserman et al. 2000], exponentially twisting the delta-gamma approximation stays, under certain conditions, asymptotically optimal for quantile estimations. More precisely, if $\lambda_{\max} > 0$ and $L = Q$, then exponential twisting using the twisting parameter defined by (29) is also asymptotically optimal for the estimation of $P(L > y_x)$, where $y_x \to \infty$ in such a way that $\limsup_{x\to\infty} \frac{x}{y_x} < \infty$.

For quantile estimation using importance sampling, we could therefore simply choose an importance sampling density designed to efficiently estimate $P(L > y)$, where y is a rough estimate of the quantile of interest. But if the first guess y is a bad quantile estimation, it may be useful to update the density during the process of simulation. Instead of generating all the simulations using the same density g, we could start with a density g_1 and choose a new one as soon as there is more information on the loss distribution available. This may result in up to N importance sampling densities $g_1, g_2, \ldots, g_N$.

We will now introduce various algorithms for the estimation of VaR_α and ES_α that combine the "direct" Monte Carlo approach from Section 3.3 with the importance sampling methods from Section 4. The desired level α for

Algo1Q$(\alpha, x_{\text{start}}, N)$
$x := x_{\text{start}}$
compute θ_x, satisfying (29)
for $j = 1 : m$
    $\mu_j(\theta_x) := \frac{\theta_x b_j}{1-2\lambda_j\theta_x}$, by (27)
    $\sigma_j^2(\theta_x) := \frac{1}{1-2\lambda_j\theta_x}$, by (28)
end
$\mu(\theta_x) := (\mu_1(\theta_x), \ldots, \mu_m(\theta_x))$
$\Sigma(\theta_x) := \text{diag}\,(\sigma_1^2(\theta_x), \ldots, \sigma_m^2(\theta_x))$
$A := N \times 2$ matrix consisting of all zeros
for $i = 1 : N$
    generate $Z \sim \mathcal{N}(\mu(\theta_x), \Sigma(\theta_x))$
    $L := f^L(Z)$, loss as in (13)
    $Q := a + b^\top Z + Z^\top \Lambda Z$, delta-gamma approximation as in (16)
    $r := \exp(-\theta_x Q + \psi(\theta_x))$, likelihood ratio as in (24)
    $A(i,:) := (L, r)$
end
sort the rows of A in descending order with respect to the first column
$k := \min\left\{j \in \{1, 2, \ldots, N\} : \frac{1}{N}\sum_{l=1}^{j} A(l,2) \geq \alpha\right\}$
$\widehat{\text{VaR}}_\alpha := A(k,1)$, by (11)
$\widehat{\text{ES}}_\alpha := \frac{1}{\alpha}\left(\frac{1}{N}\sum_{l=1}^{k-1} A(l,1)A(l,2) + \left(\alpha - \frac{1}{N}\sum_{l=1}^{k-1} A(l,2)\right) A(k,1)\right)$, by (12)

Figure 6: Algorithm for the estimation of VaR_α and ES_α using delta-gamma IS, based on a sample of N simulations. The algorithm does not update the importance sampling density throughout the simulation process. x_{start} is supposed to be a rough guess for VaR_α.

VaR and ES and a rough guess for VaR_α, called x_{start}, are input values of the estimation algorithms. That is, they have to be set in advance. Every algorithm makes use of (11) and (12) to estimate VaR_α and ES_α, respectively. The names of the algorithms contain a digit (1,2 or 3), characterizing the underlying updating scheme for the importance sampling density, and a letter (Y or Q), depending on the type of importance sampling that is applied (IS based on Y or IS based on Q). N denotes the total number of simulations that is used for the final VaR_α and ES_α estimations.

The first two algorithms, Algo1Y and Algo1Q, are presented in Figure 5 and Figure 6, respectively. Besides α and x_{start} both algorithms have N as a third input value. In Algo1Y and Algo1Q the importance sampling density is not being updated during simulation. That is, one single importance sampling density, designed for an efficient estimation of $P(L > x_{\mathrm{start}})$, is used for all the simulations. Whereas Algo1Y uses importance sampling based on Y (delta IS), Algo1Q uses exponential twisting of Q (delta-gamma IS).

Two more algorithms for estimating VaR_α and ES_α are presented in Figure 7 and Figure 8: Algo2Y and Algo2Q. Now the importance sampling density is updated on a regular basis during the simulation processs. Algo2Y and Algo2Q make use of two simple help algorithms, ProbY and ProbQ, that are described in Figure 9. Based on n simulations $\mathrm{ProbY}(x, n)$ and $\mathrm{ProbQ}(x, n)$ estimate $P(L > x)$ using delta IS and delta-gamma IS, respectively. In Algo2Y and Algo2Q the density for sampling is updated after every n simulations. This corresponds to updating the mean μ in ProbY and the parameter θ_x in ProbQ. The choice of the new importance sampling density is always based on the foregoing n simulations. More precisely, using a bisection method, we constantly compute new VaR_α estimates, based on n simulations each. A new estimate allows the selection of a new importance sampling density. Note that in Algo2Y and Algo2Q the total number of simulations N cannot be set in advance, because the bisection procedure terminates as soon as a certain accuracy level has been reached. However, as an input value, n can be altered explicitly, allowing an indirect control of N.

A method to update the importance sampling density after every single simulation is described in Algo3Y and Algo3Q, presented in Figure 10 and Figure 11, respectively. In Algo2Y and Algo2Q, the density updates were based on loss probability estimations. Now we use (11) to get a new VaR estimate after every simulation. Each new estimate in turn is used to select the importance sampling density for the next simulation. This method has two potential advantages over the approach of Algo2Y and Algo2Q. It allows more frequent density updates and the choice of every new density is based on all the foregoing simulations, resulting in more accuracy. Note that Algo3Y

Algo2Y$(\alpha, x_{\text{start}}, n)$
$A :=$ empty matrix with 2 columns
during algorithm, save each pair (L, r) from ProbY as a new row in A
$x := x_{\text{start}}$
if ProbY$(x, n) > \alpha$
    lower:$= x$
    while ProbY$(x, n) > \alpha$
        $x := 2x$
    end
    upper:$= x$
else
    upper:$= x$
    while ProbY$(x, n) \leq \alpha$
        $x := \frac{x}{2}$
    end
    lower:$= x$
end
$x := \frac{\text{upper+lower}}{2}$
while upper $-$ lower $> 1\%$ of x
    if ProbY$(x, n) > \alpha$
        lower $:= \frac{\text{upper+lower}}{2}$
    else
        upper $:= \frac{\text{upper+lower}}{2}$
    end
    $x := \frac{\text{upper+lower}}{2}$
end
$N :=$ number of rows in $A =$ total number of simulations
sort rows of A in descending order with respect to the first column
$k := \min\left\{j \in \{1, 2, \ldots, N\} : \frac{1}{N}\sum_{l=1}^{j} A(l,2) \geq \alpha\right\}$
$\widehat{\text{VaR}}_\alpha := A(k, 1)$, by (11)
$\widehat{\text{ES}}_\alpha := \frac{1}{\alpha}\left(\frac{1}{N}\sum_{l=1}^{k-1} A(l,1)A(l,2) + \left(\alpha - \frac{1}{N}\sum_{l=1}^{k-1} A(l,2)\right) A(k,1)\right)$, by (12)

Figure 7: Algorithm for the estimation of VaR_α and ES_α using delta IS. The importance sampling density is updated after every n simulations using a bisection method. x_{start} is supposed to be a rough guess for VaR_α. Algo2Y makes use of the help algorithm ProbY from Figure 9.

Algo2Q$(\alpha, x_{\text{start}}, n)$
$A :=$ empty matrix with 2 columns
during algorithm, save each pair (L, r) from ProbQ as a new row in A
$x := x_{\text{start}}$
if probQ$(x, n) > \alpha$
    lower:$= x$
    while ProbQ$(x, n) > \alpha$
        $x := 2x$
    end
    upper:$= x$
else
    upper:$= x$
    while ProbQ$(x, n) \leq \alpha$
        $x := \frac{x}{2}$
    end
    lower:$= x$
end
$x := \frac{\text{upper+lower}}{2}$
while upper $-$ lower $> 1\%$ of x
    if ProbQ$(x, n) > \alpha$
        lower $:= \frac{\text{upper+lower}}{2}$
    else
        upper $:= \frac{\text{upper+lower}}{2}$
    end
    $x := \frac{\text{upper+lower}}{2}$
end
$N :=$ number of rows in $A =$ total number of simulations
sort rows of A in descending order with respect to the first column
$k := \min\left\{j \in \{1, 2, \ldots, N\} : \frac{1}{N}\sum_{l=1}^{j} A(l, 2) \geq \alpha\right\}$
$\widehat{\text{VaR}}_\alpha := A(k, 1)$, by (11)
$\widehat{\text{ES}}_\alpha := \frac{1}{\alpha}\left(\frac{1}{N}\sum_{l=1}^{k-1} A(l, 1)A(l, 2) + \left(\alpha - \frac{1}{N}\sum_{l=1}^{k-1} A(l, 2)\right) A(k, 1)\right)$, by (12)

Figure 8: Algorithm for the estimation of VaR_α and ES_α using delta-gamma IS. The importance sampling density is updated after every n simulations using a bisection method. x_{start} is supposed to be a rough guess for VaR_α. Algo2Q makes use of the help algorithm ProbQ from Figure 9.

ProbY(x, n)
$\mu := (x - a)\frac{b}{b^\top b}$, by (21)
$p := 0$
for $i = 1 : n$
    generate $Z \sim \mathcal{N}(\mu, I)$
    $L := f^L(Z)$, by (13)
    $r := \exp\left(\frac{1}{2}\mu^\top\mu - \mu^\top Z\right)$, by (22)
    if $L > x$
        $p := p + \frac{r}{n}$, by (9)
    end
end
return p

ProbQ(x, n)
compute θ_x, satisfying (29)
for $j = 1 : m$
        $\mu_j(\theta_x) := \frac{\theta_x b_j}{1-2\lambda_j\theta_x}$, by (27)
        $\sigma_j^2(\theta_x) := \frac{1}{1-2\lambda_j\theta_x}$, by (28)
end
$\mu(\theta_x) := (\mu_1(\theta_x), \ldots, \mu_m(\theta_x))$
$\Sigma(\theta_x) := \text{diag}\,(\sigma_1^2(\theta_x), \ldots, \sigma_m^2(\theta_x))$
$p := 0$
for $i = 1 : n$
    generate $Z \sim \mathcal{N}(\mu(\theta_x), \Sigma(\theta_x))$
    $L := f^L(Z)$, by (13)
    $Q := a + b^\top Z + Z^\top \Lambda Z$, by (16)
    $r := \exp(-\theta_x Q + \psi(\theta_x))$, by (24)
    if $L > x$
        $p := p + \frac{r}{n}$, by (9)
    end
end
return p

Figure 9: ProbY and ProbQ estimate $P(L > x)$ using delta IS and delta-gamma IS, respectively. Both estimators are based on n simulations. ProbY and ProbQ are help algorithms for Algo2Y and Algo2Q, respectively.

Algo3Y$(\alpha, x_{\text{start}}, N)$
$x := x_{\text{start}}$
$A := N \times 2$ matrix consisting of all zeros
for $i = 1 : N$
    $\mu := (x - a)\frac{b}{b^\top b}$, by (21)
    generate $Z \sim \mathcal{N}(\mu, I)$
    $L := f^L(Z)$, loss as in (13)
    $r := \exp\left(\frac{1}{2}\mu^\top\mu - \mu^\top Z\right)$, likelihood ratio as in (22)
    $h := \min\{j \in \{1, 2, \ldots, i\} : L \geq A(j, 1) \text{ or } j = i\}$
    if $h = i$
        $A(h, :) := (L, r)$
    else
        $A(h + 1 : i, :) := A(h : i - 1, :)$
        $A(h, :) := (L, r)$
    end
    if $\frac{1}{i}\sum_{l=1}^{i} A(l, 2) < \alpha$
        $k := i$
    else
        $k := \min\left\{ j \in \{1, 2, \ldots, i\} : \frac{1}{i}\sum_{l=1}^{j} A(l, 2) \geq \alpha \right\}$
    end
    $x := A(k, 1)$ is new VaR_α estimate, by (11)
end
$\widehat{\text{VaR}}_\alpha := x$
$\widehat{\text{ES}}_\alpha := \frac{1}{\alpha}\left(\frac{1}{N}\sum_{l=1}^{k-1} A(l, 1)A(l, 2) + \left(\alpha - \frac{1}{N}\sum_{l=1}^{k-1} A(l, 2)\right) A(k, 1)\right)$, by (12)

Figure 10: Algorithm for the estimation of VaR_α and ES_α using delta IS, based on a sample of N simulations. The importance sampling density is updated after every simulation, using (11). x_{start} is supposed to be a rough guess for VaR_α.

Algo3Q$(\alpha, x_{\text{start}}, N)$
$x := x_{\text{start}}$
$A := N \times 2$ matrix consisting of all zeros
for $i = 1 : N$
    compute θ_x, satisfying (29)
    for $j = 1 : m$
        $\mu_j(\theta_x) := \frac{\theta_x b_j}{1-2\lambda_j\theta_x}$, by (27)
        $\sigma_j^2(\theta_x) := \frac{1}{1-2\lambda_j\theta_x}$, by (28)
    end
    $\mu(\theta_x) := (\mu_1(\theta_x), \ldots, \mu_m(\theta_x))$
    $\Sigma(\theta_x) := \text{diag}\,(\sigma_1^2(\theta_x), \ldots, \sigma_m^2(\theta_x))$
    generate $Z \sim \mathcal{N}(\mu(\theta_x), \Sigma(\theta_x))$
    $L := f^L(Z)$, loss as in (13)
    $Q := a + b^\top Z + Z^\top \Lambda Z$, delta-gamma approximation as in (16)
    $r := \exp(-\theta_x Q + \psi(\theta_x))$, likelihood ratio as in (24)
    $h := \min\{j \in \{1, 2, \ldots, i\} : L \geq A(j, 1) \text{ or } j = i\}$
    if $h = i$
        $A(h, :) := (L, r)$
    else
        $A(h + 1 : i, :) := A(h : i - 1, :)$
        $A(h, :) := (L, r)$
    end
    if $\frac{1}{i}\sum_{l=1}^{i} A(l, 2) < \alpha$
        $k := i$
    else
        $k := \min\left\{ j \in \{1, 2, \ldots, i\} : \frac{1}{i}\sum_{l=1}^{j} A(l, 2) \geq \alpha \right\}$
    end
    $x := A(k, 1)$ is new VaR_α estimate, by (11)
end
$\widehat{\text{VaR}}_\alpha := x$
$\widehat{\text{ES}}_\alpha := \frac{1}{\alpha}\left(\frac{1}{N}\sum_{l=1}^{k-1} A(l, 1)A(l, 2) + \left(\alpha - \frac{1}{N}\sum_{l=1}^{k-1} A(l, 2)\right) A(k, 1)\right)$, by (12)

Figure 11: Algorithm for the estimation of VaR_α and ES_α using delta-gamma IS, based on a sample of N simulations. The importance sampling density is updated after every simulation, using (11). x_{start} is supposed to be a rough guess for VaR_α.

uses delta importance sampling and Algo3Q is based on delta-gamma importance sampling. Both algorithms have N as a third input value, that is, the total number of simulations can be chosen in advance.

6 Numerical illustrations

In this section we will apply the algorithms from Section 5 to specific portfolios. Let us stick to the assumptions made in Section 3.1 and Section 4.1. Here we assume that there are $m = 10$ underlying assets, all having an initial value of 100 and an annual volatility of $\sigma = 0.30$. All the assets are supposed to be pairwise uncorrelated. Further, we assume 250 trading days in a year and use a continuously compounded risk-free interest rate of 5%. We study losses over a time period of ten days, that is, $t = 0.04$ years. These assumptions let us conclude that $\Delta S \sim \mathcal{N}(0, \Sigma)$, where Σ is a diagonal matrix with diagonal elements

$$\Sigma_{jj} = \left(100\sigma\sqrt{t}\right)^2 = 36,$$

for $j = 1, 2, \ldots, 10$.

We consider the following two portfolios that have also been used in [Glasserman et al. 1999] to illustrate importance sampling. Portfolio 1 consists of short positions of ten at-the-money calls on each underlying asset. Every option has an expiry of 0.5 years. In Portfolio 2 the position consists of ten short at-the-money calls and five short at-the-money puts on each asset. The options are again supposed to have a maturity of 0.5 years. For valuating the portfolios at time 0 and time t we use the Black-Scholes formula. The computation of $\frac{\partial V}{\partial t}$, $\frac{\partial V}{\partial S_i(t)}$ and $\frac{\partial^2 V}{\partial S_i(t)\partial S_j(t)}$, the portfolio sensitivities used in the Taylor expansions of L, is also based on the values provided by Black-Scholes.

Remark 6.0.1 The Black-Scholes model assumes that $S_j(t)/S_j(0)$ is lognormally distributed for $j = 1, 2, \ldots, m$. Thus, there seems to be an inconsistency between the valuation model and the model used for path simulation. But for small t

$$\frac{S_j(t)}{S_j(0)} = (1 + \Delta S_j/S_j(0)) \approx \exp(\Delta S_j/S_j(0)),$$

which is lognormally distributed if ΔS_j, the j-th component of ΔS, is normally distributed. Hence the inconsistency between the two models may be neglected here. Note that in practice it is not unusual to assume that ΔS is normally distributed, that is, pricing models are often much more detailed than the models used to describe market risk.

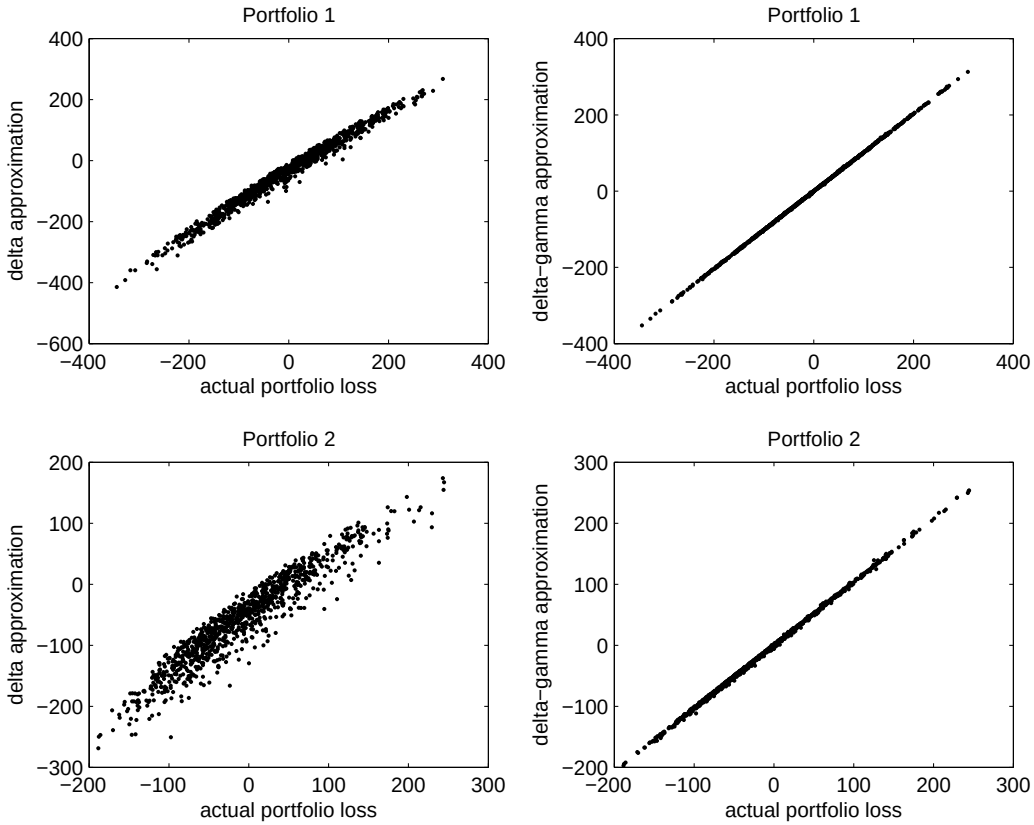

Figure 12: Comparison of actual losses of Portfolio 1 and Portfolio 2 to delta and delta-gamma approximations for 1000 simulations of ΔS.

Portfolio 1 has a strong linear component, because it only consists of call options that are quite far from expiration. This is confirmed by Figure 12, where actual losses of Portfolio 1 and Portfolio 2 are compared to first and second order Taylor approximations for 1000 simulations of ΔS. We can indeed see that for Portfolio 1 the delta approximation fits better than for Portfolio 2. The delta-gamma approximation of Portfolio 2 seems to be very accurate. Hence Figure 12 suggests to use delta IS for Portfolio 1 and delta-gamma IS for Portfolio 2. That is, we will compare the performances of Algo1Y, Algo2Y and Algo3Y using Portfolio 1 and Algo1Q, Algo2Q and Algo3Q will be applied to Portfolio 2. The two portfolios that are used here are quite simple and do of course not ensure an exhaustive comparison, but they should be sufficient to roughly illustrate how the presented algorithms perform in practice.

A first step in estimating risk measures is to make a rough guess for VaR_α

Portfolio 1	$\alpha = 0.01\%$	$\alpha = 0.1\%$	$\alpha = 1\%$	$\alpha = 5\%$
$\mathrm{VaR}_\alpha^{\mathrm{true}}$	442.16	361.09	262.63	178.36
$\mathrm{VaR}_\alpha^{\delta}$	372.47	302.25	216.94	140.83
Portfolio 2	$\alpha = 0.01\%$	$\alpha = 0.1\%$	$\alpha = 1\%$	$\alpha = 5\%$
$\mathrm{VaR}_\alpha^{\mathrm{true}}$	322.91	259.49	185.06	123.24
$\mathrm{VaR}_\alpha^{\delta\text{-}\Gamma}$	338.44	270.10	192.27	127.63

Table 1: Comparison of $(1-\alpha)$-quantiles of the delta and delta-gamma approximations ($\mathrm{VaR}_\alpha^{\delta}$ and $\mathrm{VaR}_\alpha^{\delta\text{-}\Gamma}$) with "true" values for VaR_α, that is, standard Monte Carlo estimates based on a sample of $2 \cdot 10^6$ simulations.

that can be used as x_{start} in the algorithms from Section 5. According to Figure 12, it would make sense to use the delta and delta-gamma approximations to make a first quantile estimation. Using (18), we can easily compute quantiles of the delta approximation Y. The function *optimize()* from the statistics software *R 2.10.0* allows to numerically compute a solution of

$$P(R \leq x) - p = 0,$$

for $p \in (0,1)$ and $P(R \leq x)$ given by (20). This in turn provides us with quantiles of Q. Table 1 compares so-called "true" values of VaR_α to $(1-\alpha)$-quantiles of the delta and delta-gamma approximations ($\mathrm{VaR}_\alpha^{\delta}$ and $\mathrm{VaR}_\alpha^{\delta\text{-}\Gamma}$), for some commonly used values of α. Each $\mathrm{VaR}_\alpha^{\mathrm{true}}$ has been computed by standard Monte Carlo (using (7)) based on a huge sample of $2 \cdot 10^6$ simulations. The quantile estimates provided by Y and Q may not be accurate enough to serve as risk measures, but as mentioned above, they can at least be used as start values in our algorithms.

Let us now apply Algo1Y, Algo2Y and Algo3Y to Portfolio 1 and analyze their performances. Recall that these three algorithms use delta IS. Algo1Y selects one importance sampling density and sticks to it. Algo2Y updates the density after every n simulations and in Algo3Y the update takes place after each simulation. First we will study the sensitivities of the algorithms to x_{start}. Table 2 presents the results for $\alpha = 5\%$. The data corresponding to $\alpha = 1\%$ are represented in Table 3. The first column indicates the start values that have been used. Among the values for x_{start} are very accurate VaR_α estimates, but the behavior of the algorithms is also analyzed for "bad" start values. Note that $\mathrm{VaR}_\alpha^{\mathrm{true}}$ and $\mathrm{VaR}_\alpha^{\delta}$ are taken from Table 1. For each value of x_{start}, every algorithm has been run 100 times, each run being based on a sample of size N. For every algorithm mean and standard deviation of the resulting 100 VaR_α and ES_α estimates have been reported. In the tables the standard deviation is placed below the corresponding mean, between

x_{start}	N	Algo1Y		Algo2Y		Algo3Y	
		$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$	$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$	$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$
$25\% \cdot \text{VaR}_\alpha^{\text{true}}$	532	177.61	228.54	177.31	229.10	177.99	229.44
		(5.52)	(5.76)	(4.23)	(2.57)	(3.81)	(2.62)
$50\% \cdot \text{VaR}_\alpha^{\text{true}}$	488	178.54	229.94	178.24	230.31	178.28	229.65
		(4.42)	(3.39)	(4.48)	(2.38)	(3.94)	(2.66)
$75\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	178.09	229.51	177.37	228.82	177.47	229.39
		(4.28)	(2.81)	(4.33)	(2.69)	(3.81)	(2.58)
$\text{VaR}_\alpha^{\delta}$	478	177.62	229.34	177.67	229.86	177.61	229.29
		(3.87)	(2.57)	(3.97)	(2.61)	(3.53)	(2.44)
$\text{VaR}_\alpha^{\text{true}}$	474	177.70	229.86	177.42	229.52	178.37	229.46
		(4.00)	(2.35)	(4.45)	(2.85)	(4.06)	(2.20)
$125\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	177.82	229.43	177.52	229.63	177.15	229.07
		(4.16)	(2.00)	(3.66)	(2.56)	(3.66)	(2.38)
$150\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	177.36	229.18	177.59	229.29	178.31	229.42
		(5.31)	(2.70)	(3.77)	(2.45)	(4.64)	(2.56)
$175\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	177.03	228.80	177.39	229.43	177.34	228.89
		(7.18)	(3.83)	(4.69)	(2.40)	(4.79)	(2.54)

Table 2: Portfolio 1, $\alpha = 5\%$. Sensitivities to x_{start} of Algo1Y, Algo2Y and Algo3Y. Means and standard deviations (between brackets) of 100 algorithm runs are reported. We assume that $\text{VaR}_\alpha^{\text{true}} = 178.36$, $\text{ES}_\alpha^{\text{true}} = 230.08$ and $\text{VaR}_\alpha^{\delta} = 140.83$. Standard Monte Carlo results for $N = 500$ and 100 runs: $\widehat{\text{VaR}}_\alpha = 176.91$ (12.31), $\widehat{\text{ES}}_\alpha = 228.77$ (13.38).

x_{start}	N	Algo1Y		Algo2Y		Algo3Y	
		$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$	$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$	$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$
$25\% \cdot \text{VaR}_\alpha^{\text{true}}$	528	261.18	303.92	261.66	304.43	261.63	304.65
		(7.42)	(6.91)	(5.06)	(3.43)	(3.53)	(2.52)
$50\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	262.47	305.22	262.53	304.76	261.87	304.65
		(4.52)	(3.90)	(4.61)	(2.71)	(3.89)	(2.17)
$75\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	261.42	304.47	262.48	305.12	261.45	304.81
		(4.14)	(2.71)	(4.59)	(2.64)	(3.84)	(2.45)
VaR_α^δ	480	262.22	304.95	261.49	304.69	261.63	305.06
		(3.89)	(2.32)	(4.29)	(2.56)	(4.20)	(2.39)
$\text{VaR}_\alpha^{\text{true}}$	469	261.97	304.78	261.65	304.95	261.68	304.86
		(4.14)	(2.21)	(4.88)	(2.80)	(3.77)	(2.36)
$125\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	261.94	304.99	262.01	305.19	261.83	304.86
		(5.36)	(2.47)	(3.53)	(2.42)	(4.22)	(2.48)
$150\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	260.90	304.48	261.18	304.61	261.29	304.95
		(6.36)	(3.92)	(4.33)	(2.39)	(3.86)	(2.43)
$175\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	259.57	303.56	261.00	305.10	261.45	304.70
		(14.13)	(6.66)	(4.87)	(2.97)	(4.22)	(2.74)

Table 3: Portfolio 1, $\alpha = 1\%$. Sensitivities to x_{start} of Algo1Y, Algo2Y and Algo3Y. Means and standard deviations (between brackets) of 100 algorithm runs are reported. We assume that $\text{VaR}_\alpha^{\text{true}} = 262.63$, $\text{ES}_\alpha^{\text{true}} = 305.67$ and $\text{VaR}_\alpha^\delta = 216.94$. Standard Monte Carlo results for $N = 500$ and 100 runs: $\widehat{\text{VaR}}_\alpha = 257.07$ (19.00), $\widehat{\text{ES}}_\alpha = 298.75$ (27.08).

x_{start}	N	Algo1Q		Algo2Q		Algo3Q	
		$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$	$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$	$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$
$25\% \cdot \text{VaR}_\alpha^{\text{true}}$	528	123.88	161.53	123.61	161.47	123.18	161.41
		(4.48)	(5.15)	(2.98)	(2.58)	(2.83)	(2.07)
$50\% \cdot \text{VaR}_\alpha^{\text{true}}$	495	123.13	160.91	123.26	161.58	123.42	161.07
		(4.26)	(3.84)	(2.84)	(2.26)	(2.86)	(2.10)
$75\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	123.56	161.78	122.98	161.42	122.93	161.30
		(3.22)	(2.53)	(2.98)	(2.01)	(2.97)	(2.17)
$\text{VaR}_\alpha^{\text{true}}$	472	123.17	161.09	123.14	161.23	123.99	161.45
		(2.72)	(2.05)	(2.94)	(2.24)	(2.74)	(1.82)
$\text{VaR}_\alpha^{\delta\text{-}\Gamma}$	476	123.61	161.58	123.72	161.57	123.31	161.55
		(3.22)	(2.01)	(2.97)	(2.34)	(3.03)	(2.28)
$125\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	123.27	161.33	122.97	161.08	122.83	161.20
		(3.02)	(2.15)	(3.30)	(2.26)	(3.06)	(2.20)
$150\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	123.30	161.45	123.17	161.16	123.67	161.48
		(2.84)	(1.78)	(3.00)	(2.26)	(3.28)	(2.01)
$200\% \cdot \text{VaR}_\alpha^{\text{true}}$	494	123.45	161.37	123.04	161.35	123.56	161.19
		(3.71)	(1.84)	(3.25)	(2.26)	(3.34)	(2.04)
$250\% \cdot \text{VaR}_\alpha^{\text{true}}$	520	123.11	161.15	123.58	161.54	123.11	161.19
		(4.63)	(2.67)	(2.99)	(2.20)	(3.18)	(2.21)
$300\% \cdot \text{VaR}_\alpha^{\text{true}}$	520	122.53	161.24	123.68	161.56	123.16	161.07
		(7.70)	(3.43)	(3.43)	(2.26)	(3.11)	(2.09)

Table 4: Portfolio 2, $\alpha = 5\%$. Sensitivities to x_{start} of Algo1Q, Algo2Q and Algo3Q. Means and standard deviations (between brackets) of 100 algorithm runs are reported. We assume that $\text{VaR}_\alpha^{\text{true}} = 123.24$, $\text{ES}_\alpha^{\text{true}} = 161.22$ and $\text{VaR}_\alpha^{\delta\text{-}\Gamma} = 127.63$. Standard Monte Carlo results for $N = 500$ and 100 runs: $\widehat{\text{VaR}}_\alpha = 123.63$ (8.37), $\widehat{\text{ES}}_\alpha = 162.05$ (9.88).

x_{start}	N	Algo1Q		Algo2Q		Algo3Q	
		$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$	$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$	$\widehat{\text{VaR}}_\alpha$	$\widehat{\text{ES}}_\alpha$
$25\% \cdot \text{VaR}_\alpha^{\text{true}}$	525	185.77	218.77	185.45	218.09	185.54	218.20
		(7.00)	(7.55)	(3.67)	(2.83)	(2.81)	(2.28)
$50\% \cdot \text{VaR}_\alpha^{\text{true}}$	487	185.42	217.76	185.34	218.06	185.82	217.79
		(4.03)	(3.96)	(2.96)	(2.33)	(3.18)	(2.25)
$75\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	185.96	218.50	185.06	217.82	185.48	218.19
		(3.19)	(2.39)	(2.79)	(2.09)	(2.63)	(2.17)
$\text{VaR}_\alpha^{\text{true}}$	470	185.40	217.83	185.67	218.55	185.10	218.04
		(2.58)	(2.05)	(3.02)	(2.16)	(2.66)	(1.90)
$\text{VaR}_\alpha^{\delta\text{-}\Gamma}$	477	185.20	218.17	185.36	218.11	185.21	217.42
		(2.96)	(1.87)	(2.64)	(1.93)	(3.01)	(2.06)
$125\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	184.82	217.86	185.37	217.91	185.36	218.90
		(2.29)	(2.13)	(3.28)	(2.48)	(3.08)	(2.08)
$150\% \cdot \text{VaR}_\alpha^{\text{true}}$	480	184.67	217.77	184.80	217.81	184.93	218.09
		(3.00)	(1.72)	(3.18)	(1.88)	(2.40)	(1.88)
$200\% \cdot \text{VaR}_\alpha^{\text{true}}$	490	185.05	217.80	185.59	218.23	185.23	218.17
		(4.83)	(2.34)	(2.77)	(1.98)	(3.26)	(2.14)
$250\% \cdot \text{VaR}_\alpha^{\text{true}}$	520	184.54	217.48	185.13	217.90	185.28	218.26
		(7.34)	(3.94)	(2.97)	(2.16)	(3.00)	(2.07)

Table 5: Portfolio 2, $\alpha = 1\%$. Sensitivities to x_{start} of Algo1Q, Algo2Q and Algo3Q. Means and standard deviations (between brackets) of 100 algorithm runs are reported. We assume that $\text{VaR}_\alpha^{\text{true}} = 185.06$, $\text{ES}_\alpha^{\text{true}} = 217.65$ and $\text{VaR}_\alpha^{\delta\text{-}\Gamma} = 192.27$. Standard Monte Carlo results for $N = 500$ and 100 runs: $\widehat{\text{VaR}}_\alpha = 182.19$ (14.46), $\widehat{\text{ES}}_\alpha = 215.50$ (19.97).

brackets. Remember that for Algo2Y it is not possible to set N in advance, because it depends on x_{start} and n. Here, we set $n = 40$ for Algo2Y, resulting in a sample size of about 500. The number N indicated in the tables is the average number of simulations used by Algo2Y in 100 runs. Every run of Algo1Y and Algo3Y is based on exactly N simulations. $\text{ES}_\alpha^{\text{true}}$, corresponding to a standard Monte Carlo estimation based on a sample of size $2 \cdot 10^6$, is indicated in the captions of Table 2 and Table 3. Note that all the estimations have been made using *MATLAB R2009a*.

Table 4 and Table 5 represent the sensitivities of Algo1Q, Algo2Q and Algo3Q to x_{start} for $\alpha = 5\%$ and $\alpha = 1\%$, respectively. Since these algorithms are based on delta-gamma IS, they have been applied to Portfolio 2. Again, the values of $\text{VaR}_\alpha^{\text{true}}$ and $\text{VaR}_\alpha^{\delta\text{-}\Gamma}$ are taken from Table 1 and $\text{ES}_\alpha^{\text{true}}$, indicated in the caption of each table, is a standard Monte Carlo estimation based on $2 \cdot 10^6$ simulations. As before, the reported results are means and standard deviations of 100 algorithm runs. In Algo2Q, n has been set to 40, that is, for each run of Algo2Q the importance sampling density is updated after every 40 simulations.

How to interpret the data reported in Tables 2, 3, 4 and 5? Let us first compare the standard deviations of the importance sampling estimates to those of the corresponding standard Monte Carlo estimations. The results of standard Monte Carlo, for $N = 500$, are reported in the captions of the tables. We can see that, independently of algorithm and start value, importance sampling leads to dramatic variance reduction and hence computational speed-up, especially for small values of α. More precisely, for $\alpha = 1\%$, making use of IS to estimate VaR reduces the standard deviation by a factor of about 5, if a suitable start value is chosen. For the estimation of ES, the corresponding standard deviation ratios are approximately equal to 10. Assuming roughly equal computing times per sample with and without IS, we can conclude that an importance sampling estimator produces as precise an estimate for VaR_α (ES_α) as standard Monte Carlo in $1/5^2$ ($1/10^2$) as much computing time. Let us now analyze the performances of the different importance sampling algorithms. If $\text{VaR}_\alpha^{\delta}$ or $\text{VaR}_\alpha^{\delta\text{-}\Gamma}$ are chosen as start values, all the algorithms provide quite accurate estimations. That is, in this case updating the importance sampling density seems to be unnecessary. The results of Section 5, stating that delta-gamma IS is not sensitive to the choice of a good first guess, are confirmed by our results. Algo1Q makes good estimations for a quite broad range of start values. However, delta IS seems to be more sensitive to the choice of x_{start}. Even for small deviations of x_{start} from $\text{VaR}_\alpha^{\text{true}}$, the results of Algo1Y get slightly worse. The standard deviations of the estimations made by Algo2Y, Algo3Y, Algo2Q and Algo3Q are almost the same for all the start values. This is not the case for Algo1Y

n	N	Algo2Y		Algo3Y		sd ratios	
		$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$	$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$	$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$
10	132	259.08	303.35	261.31	304.81		
		(8.12)	(6.31)	(7.75)	(4.69)	(1.05)	(1.35)
25	335	261.50	304.53	261.93	304.82		
		(5.50)	(4.01)	(4.72)	(2.71)	(1.17)	(1.48)
50	660	261.12	304.56	261.63	305.24		
		(3.76)	(2.83)	(3.60)	(2.19)	(1.04)	(1.29)
75	990	261.35	304.75	262.30	305.01		
		(4.68)	(2.88)	(2.65)	(1.53)	(1.77)	(1.88)
100	1304	261.84	305.00	261.53	304.73		
		(3.39)	(2.18)	(2.30)	(1.46)	(1.47)	(1.49)
150	1960	261.62	304.77	262.20	305.12		
		(2.84)	(1.79)	(2.03)	(1.26)	(1.40)	(1.42)

Table 6: Portfolio 1, $\alpha = 1\%$. Sensitivity of Algo2Y to update frequency. Varying n for fixed start value $x_{\text{start}} = 25\% \cdot \mathrm{VaR}_\alpha^{\text{true}}$. Means and standard deviations (between brackets) of 100 algorithm runs are reported. The last two columns contain the standard deviation ratios of the VaR_α and ES_α estimates, respectively. $\mathrm{VaR}_\alpha^{\text{true}} = 262.63$ and $\mathrm{ES}_\alpha^{\text{true}} = 305.67$.

and Algo1Q, where standard deviations increase considerably if x_{start} is getting worse. Note that ES estimations are much more sensitive to small start values than to big ones. For the estimation of VaR, the change of standard deviation is of the same magnitude for both, over- and underestimated values of x_{start}. This is not surprising, because for the estimation of expected shortfall much more simulations in the tail of the distribution are needed. Furthermore we can see that for $\alpha = 1\%$ Algo1Y and Algo1Q are much more sensitive to high values of x_{start} than for $\alpha = 5\%$. All this shows that, if there is no mean to get a good first guess of VaR_α, regular updates of the importance sampling density during the simulation process can be useful. But which is the most efficient updating scheme? If we compare Algo2Y to Algo3Y and Algo2Q to Algo3Q, we cannot see big differences in the standard deviations. But all in all, Algo3Y and Algo3Q seem to provide slightly more accurate results than Algo2Y and Algo2Q, respectively.

To find out whether there is a possibility to make Algo2Y and Algo2Q more accurate we will vary n for fixed start values and compare the results to those of Algo3Y and Algo3Q. This will allow to see if it is more efficient to choose n rather small or rather large. Especially for "bad" start values it seems intuitively appealing to select a small n, allowing to get a

n	N	Algo2Y		Algo3Y		sd ratios	
		$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$	$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$	$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$
10	121	260.82	303.86	257.93	301.07		
		(9.41)	(5.40)	(12.16)	(8.98)	(0.77)	(0.60)
25	300	261.22	304.68	260.81	304.07		
		(6.33)	(3.26)	(5.01)	(3.14)	(1.26)	(1.04)
50	600	262.30	305.06	262.10	305.06		
		(5.10)	(2.69)	(4.26)	(3.14)	(1.20)	(0.86)
75	900	262.55	305.30	262.28	305.09		
		(3.99)	(2.19)	(5.38)	(4.60)	(0.74)	(0.48)
100	1200	261.80	304.85	262.33	305.20		
		(2.99)	(1.91)	(2.79)	(1.87)	(1.07)	(1.02)
150	1800	261.20	304.82	261.86	305.03		
		(2.61)	(1.61)	(1.89)	(1.21)	(1.38)	(1.33)

Table 7: Portfolio 1, $\alpha = 1\%$. Sensitivity of Algo2Y to update frequency. Varying n for fixed start value $x_{\text{start}} = 175\% \cdot \mathrm{VaR}_\alpha^{\text{true}}$. Means and standard deviations (between brackets) of 100 algorithm runs are reported. The last two columns contain the standard deviation ratios of the VaR_α and ES_α estimates, respectively. $\mathrm{VaR}_\alpha^{\text{true}} = 262.63$ and $\mathrm{ES}_\alpha^{\text{true}} = 305.67$.

n	N	Algo2Q		Algo3Q		sd ratios	
		$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$	$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$	$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$
10	133	183.99	216.23	183.78	216.56		
		(7.72)	(5.84)	(6.48)	(4.79)	(1.19)	(1.22)
25	330	185.19	218.09	185.04	217.70		
		(4.92)	(4.62)	(3.69)	(2.82)	(1.33)	(1.64)
50	660	185.13	218.14	185.73	218.00		
		(3.10)	(2.48)	(2.68)	(2.09)	(1.16)	(1.19)
75	990	185.34	217.99	185.02	217.75		
		(2.48)	(1.96)	(2.21)	(1.55)	(1.12)	(1.26)
100	1310	185.59	218.06	185.26	217.83		
		(2.09)	(1.57)	(1.97)	(1.06)	(1.06)	(1.48)
150	1950	184.90	217.78	185.38	217.97		
		(1.65)	(1.51)	(1.60)	(0.94)	(1.03)	(1.61)

Table 8: Portfolio 2, $\alpha = 1\%$. Sensitivity of Algo2Q to update frequency. Varying n for fixed start value $x_{\text{start}} = 25\% \cdot \mathrm{VaR}_\alpha^{\text{true}}$. Means and standard deviations (between brackets) of 100 algorithm runs are reported. The last two columns contain the standard deviation ratios of the VaR_α and ES_α estimates, respectively. $\mathrm{VaR}_\alpha^{\text{true}} = 185.06$ and $\mathrm{ES}_\alpha^{\text{true}} = 217.65$.

n	N	Algo2Q		Algo3Q		sd ratios	
		$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$	$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$	$\widehat{\mathrm{VaR}}_\alpha$	$\widehat{\mathrm{ES}}_\alpha$
10	120	185.46	217.52	185.14	217.47		
		(6.11)	(4.82)	(6.43)	(4.18)	(0.95)	(1.15)
25	300	185.20	218.07	185.71	217.79		
		(3.75)	(2.76)	(3.93)	(2.97)	(0.95)	(0.93)
50	600	184.96	217.91	185.70	218.15		
		(2.81)	(1.94)	(2.86)	(1.85)	(0.98)	(1.05)
75	900	185.45	218.00	185.21	217.88		
		(2.36)	(1.44)	(2.01)	(1.49)	(1.17)	(0.97)
100	1200	185.00	217.76	185.30	217.84		
		(1.83)	(1.23)	(2.02)	(1.30)	(0.91)	(0.95)
150	1800	185.52	218.10	185.34	217.92		
		(1.50)	(1.10)	(1.43)	(0.94)	(1.05)	(1.17)

Table 9: Portfolio 2, $\alpha = 1\%$. Sensitivity of Algo2Q to update frequency. Varying n for fixed start value $x_{\text{start}} = 150\% \cdot \mathrm{VaR}_\alpha^{\text{true}}$. Means and standard deviations (between brackets) of 100 algorithm runs are reported. The last two columns contain the standard deviation ratios of the VaR_α and ES_α estimates, respectively. $\mathrm{VaR}_\alpha^{\text{true}} = 185.06$ and $\mathrm{ES}_\alpha^{\text{true}} = 217.65$.

more efficient density soon. But choosing a large value for n provides more accurate intermediate estimations and thus more suitable densities. Let us consider the case where $\alpha = 1\%$. Table 6 compares Algo2Y and Algo3Y for varying n and $x_{\text{start}} = 25\% \cdot \mathrm{VaR}_\alpha^{\text{true}}$. The results in Table 7 correspond to $x_{\text{start}} = 175\% \cdot \mathrm{VaR}_\alpha^{\text{true}}$. As before, mean and standard deviation of 100 algorithm runs are reported in the tables. Additionally, to ensure a suitable comparison of Algo2Y and Algo3Y, the last two columns of Table 6 and Table 7 contain the standard deviation ratios of the VaR and ES estimates, respectively. Small ratios correspond to accurate Algo2Y estimations. Table 3 shows that, for $n = 40$, Algo3Y outperforms Algo2Y for both start values that have been chosen here. Indeed, the standard deviations of the Algo2Y results are considerably higher than those of the Algo3Y estimations. Table 8 and Table 9 present comparisons of Algo2Q and Algo3Q for two different start values, $x_{\text{start}} = 25\% \cdot \mathrm{VaR}_\alpha^{\text{true}}$ and $x_{\text{start}} = 150\% \cdot \mathrm{VaR}_\alpha^{\text{true}}$. Again, for $n = 40$, Algo2Q has provided rather poor results for both start values, as can be seen in Table 5. Let us now try to interpret the results in Tables 6, 7, 8 and 9. According to the VaR estimates in Table 8, accurate intermediate estimations seem to be more important than a frequent update of the importance sampling density, because the results of Algo2Q get more

accurate for n getting larger. But Table 6 and Table 7 suggest to choose n rather small or intermediate. Table 9 does not clearly privilege neither large nor small values of n. Thus, our results show that the performances of Algo2Y and Algo2Q, compared to those of Algo3Y and Algo3Q, depend on n, but it is not clear whether it is more suitable to choose large or small values for n.

To conclude this section let us recall that using importance sampling to estimate Value-at-Risk and expected shortfall can reduce variance considerably. In our specific case of delta and delta-gamma importance sampling, variance reduction carries over from loss probability estimation to quantile estimation. Additionally it is numerically confirmed that, if there is no easy way to get a good first guess for the quantile of interest, updating the importance sampling density during simulation increases sampling efficiency. However, the numerical examples did not allow to clearly determine the best updating scheme. On the one hand, for most start values updating after each simulation using (11) might provide slightly more accurate results than using a bisection method based on probability estimations. On the other hand, selecting N new densities may need a lot of computational effort. For delta IS the differences in computing times between standard Monte Carlo, Algo1Y, Algo2Y and Algo3Y are insignificant. But for delta-gamma IS the computation of N twisting parameters is indeed quite time consuming. Note that a run of Algo3Q takes in average 1.7 times longer than a run of Algo2Q. The most efficient updating scheme is therefore probably a compromise between the two methods presented in this thesis. That is, one could try to alter Algo3Y and Algo3Q in such a way that the IS density is not updated after every single simulation.

References

[Bucklew 1990] Bucklew, J. A., 1990. Large Deviation Techniques in Decision, Simulation and Estimation. John Wiley & Sons, Inc., New York.

[Chen et al. 1993] Chen, J.-C., Lu, D., Sadowsky, J., Yao., K., 1993. On importance sampling in digital communications-Part I: Fundamentals. IEEE J. Selected Areas Comm. 11, 289-299.

[Gil-Palaez 1951] Gil-Pelaez, J., 1951. Note on the inversion theorem. Biometrika 38, 481-482.

[Glasserman et al. 1999] Glasserman, P., Heidelberger, P. and Shahabuddin, P., 1999. Importance sampling and stratification for Value-at-Risk. Com-

putational Finance 1999 (Proceedings of the Sixth International Conference on Computational Finance) 7-24. Y.S. Abu-Mostafa, B. LeBaron, A.W. Lo, A.S. Weigend, eds., MIT Press, Cambridge, Mass.

[Glasserman et al. 2000] Glasserman, P., Heidelberger, P. and Shahabuddin, P., 2000. Variance reduction techniques for estimating value-at-risk. Management Science 46, 1349-1364.

[Glasserman et al. 2002] Glasserman, P., Heidelberger, P. and Shahabuddin, P., 2002. Portfolio Value-at-Risk with heavy-tailed risk factors. Math. Financ. 12, 239-270.

[Glasserman 2003] Glasserman, P., 2003. Monte Carlo Metods in Financial Engineering. Springer, New York.

[Glynn 1996] Glynn, P., 1996. Importance sampling for Monte Carlo estimation of quantiles. Mathematical Methods in Stochastic Simulation and Experimental Design: Proceedings of the 2nd St. Petersburg Workshop on Simulation 180-185. Publishing House of Saint Petersburg University.

[Gurland 1948] Gurland, J., 1948. Inversion formulae for the distribution of ratios. Ann. Math. Statist. 19, 228-37.

[Imhof 1961] Imhof, J., 1961. Computing the distribution of quadratic forms in normal variables. Biometrika 48, 419-426.

[Johnson et al. 1995] Johnson, N. L., Kotz, S., Balakrishnan, N., 1995. Continuous Univariate Distributions, Volume 2, Second Edition. Wiley, New York.