Acceleration of multi-factor Merton model Monte Carlo simulation via Importance Sampling and GPU parallelization

M. Béreš & R. Briš
VŠB-Technical University of Ostrava, Ostrava, Czech Republic

ABSTRACT: Credit risk refers to the risk of losses due to unexpected credit events, as a default of a counterparty. The modelling and controlling of credit risk is a very important topic within banks. Very popular and frequently used tools for modelling credit risk are multi-factor Merton models. Practical implementation of these models requires time-consuming Monte Carlo (MC) simulations, which significantly limits their usability in daily credit risk calculation. In this paper we present acceleration techniques of Merton model Monte Carlo simulations, concretely parallel GPU implementation and importance sampling (IS) employment. As the importance sampling distribution we choose the Gaussian mixture model and for calculating the IS shifted probability distribution we use the cross-entropy (CE) method. The speed-up results are demonstrated using portfolio value at risk (VaR) and expected shortfall (ES) calculation.

1 INTRODUCTION

In this paper we present a new approach to the importance sampling (IS) in the multi-factor Merton model. In the standard IS approach the normal distribution is used as a family of the IS distributions. This approach results in a decent variance reduction but a certain level of degeneracy of probability can be observed. The observed degeneracy of probability is caused by a relatively high difference between the IS distribution chosen from the normal distribution family and the optimal IS distribution and it also limits the achievable variance reduction. As a correction to this problem we use the Gaussian mixture model for the IS family of distributions. This new approach limits the level of the observed degeneracy of probability as well as increases the variance reduction.

The other significant part of this paper is the implementation of discussed models and IS procedures via CUDA on the the GPU devices. The GPU implementation of the model enables very fast calculation of the observed parameters (VaR or ES) with or without the use of the IS.

First we present a short recapitulation of the multi-factor Merton model and the terminology used, then we state a detailed specification of the tested model. For a deeper understanding of the Merton model see (Lütkebohmert 2008).

1.1 *Briefly about multi-factor Merton model*

Let assume we have a portfolio of N risky loans (*exposures*) indexed by $n = 1, \dots, N$. We are interested in the possible defaults, which can occur in the fixed time interval $[0, T]$. Let D_n denote the default indicator of an exposure n, which can be represented as a Bernoulli random variable taking the values

$$D_n = \begin{cases} 1, & \text{if the exposure } n \text{ is in the default} \\ 0, & \text{otherwise} \end{cases} . \quad (1)$$

We assume that the probabilities $PD_n = \mathbb{P}(D_n = 1)$ are given as a portfolio parameter.

The portion of the exposure n which can be lost in the time of default is called the exposure at default denoted by EAD_n. For simplicity we assume EAD_n is constant in the whole time interval $[0, T]$ and it is given as the portfolio parameter.

The portion of EAD_n representing the real loss in the case of default, is given by a random variable *loss given at default* $LGD_n \in [0, 1]$. The distribution, the expectation $ELGD_n$ and the standard deviation $VLGD_n$ of LGD_n are given as the portfolio parameters. The portfolio loss L_N is than defined as a random variable

$$L_N = \sum_{n=1}^{N} EAD_n \cdot LGD_n \cdot D_n. \quad (2)$$

Now we can define the value at risk (*VaR*) as p

quantile (or *confidence level*) of L_N

$$\begin{aligned} VaR_p(L_N) &= \inf\{x \in \mathbb{R} : \mathbb{P}(L_N > x) \leq 1-p\} \\ &= \inf\{x \in \mathbb{R} : F_{L_N}(x) \geq p\}, \end{aligned} \tag{3}$$

where $F_{L_N}(x)$ is the cumulative distribution function of L_N. And the expected shortfall (*ES*) as a conditional tail expectation with the condition $x \geq VaR_p(L_N)$

$$\begin{aligned} ES_p(L_N) &= \frac{1}{1-p}\int_{VaR_p(L_N)}^{\infty} x\,\mathbb{P}(L_N = x)\,dx \\ &= \frac{1}{1-p}\int_p^1 VaR_u(L_N)\,du. \end{aligned} \tag{4}$$

1.1.1 *Exposure correlation factors*

In the reasonable portfolio the single exposure's defaults are correlated, let us outline, how the correlation is handled in the Merton model. We assume that every exposure has a unique owner (obligor). Let $V_n(t)$ denote n-th obligor's assets, $S_n(t)$ obligor n equity and $B_n(t)$ obligor n bond, so

$$V_n(t) = S_n(t) + B_n(t), 0 \leq t \leq T. \tag{5}$$

In the Merton model a default can occur only at the maturity T, which leads into two possibilities

1. $V_n(T) > B_n(T)$: obligor has sufficient asset to fulfil debt, $D_n = 0$
2. $V_n(T) \leq B_n(T)$: obligor cannot fulfil debt and defaults, $D_n = 1$

Let r_n denote the n-th obligor's asset-value log-return $r_n = \log(V_n(T)/V_n(0))$. The multi-factor Merton model assumptions to resolve correlations between exposure defaults are:

1. r_n depends linearly on K standard normally distributed risk (*systemic*) factors $X = (X_1, \ldots, X_K)$
2. r_n depends linearly on the standard normally distributed idiosyncratic term ε_n, which is independent of the systemic factors X_k
3. single idiosyncratic factors ε_n are uncorrelated
4. asset-value log-return random variable can be represented as $r_n = \beta_n \cdot Y_n + \sqrt{1-\beta_n^2}\cdot\varepsilon_n$, where $Y_n = \sum\limits_{k=1}^{K}\alpha_{n,k}X_k$ represents exposure composite factor, β_n represents exposure sensitivity to systemic risk and weights $\alpha_{n,k}$ represents dependence on single factors X_k
5. r_n has standard normal distribution if condition $\sum\limits_{k=1}^{K}\alpha_{n,k}^2 = 1$ is satisfied

Variables $\alpha_{n,k}$ and β_n are assumed as a given portfolio parameters.

When PD_n is given and r_n has the standard normal distribution, one can calculate threshold $c_n = \Phi^{-1}(1-PD_n)$ so default indicator can be represented as

$$D_n = r_n > c_n. \tag{6}$$

1.1.2 *Monte Carlo simulation of multi-factor Merton model*

With previous knowledge and full portfolio specification we can now approximate the portfolio VaR and ES via the Monte Carlo simulations. Single exposure defaults can be directly calculated from the systemic and the idiosyncratic shocks $X_k^{(i)}$ and $\varepsilon_n^{(i)}$ drawn from the standard normal distribution $N(0,1)$, upper index (i) indicate index of the Monte Carlo sample. With the generated random $LGD_n^{(i)}$ we can calculate the total random scenario loss

$$L_N^{(i)} = \sum_{n=1}^{N} EAD_n \cdot LGD_n^{(i)} \cdot D_n^{(i)}. \tag{7}$$

The Monte Carlo simulation consisting of M trials approximate portfolio VaR as $\overline{VaR_p}(L_N) = \min\left\{L_N^{(i)} : \psi(L_N^{(i)}) \leq (1-p)\cdot M\right\} =$

$$= L_N^{[\lceil M\cdot p\rceil]}, \tag{8}$$

where $\psi\left(L_N^{(i)}\right) = \sum\limits_{j=1}^{M}(L_N^{(j)} > L_N^{(i)})$, $L_N^{[j]}$ is the j-th loss in the ascendant sorted loss sequence $L_N^{(i)}$, and ES as

$$\overline{ES_p}(L_N) = \frac{1}{M - \lceil M\cdot p\rceil}\cdot\sum_{j=\lceil M\cdot p\rceil}^{M} L_N^{[j]}. \tag{9}$$

1.2 *Tested portfolio structure specification*

The most important part of the multi-factor Merton model is the structure of the portfolio (exposure dependence on the risk factors). To obtain a portfolio with a realistic behaviour we use a natural risk factor construction considering the region-industry (*sector*) and the direct (*hierarchy*) links between exposures.

Hierarchy links are represented by hierarchy systemic factors (HSF), which can be interpreted as direct links between the exposures (for example two subsidiary companies with a common parent company), each of these systemic factors usually has impact only on a small fraction of the portfolio exposures. Sector links are represented by sector systemic factors (SSF), which can be interpreted as industrial and regional factors, each of these systemic factors usually impacts majority of the portfolio exposures. Therefore every exposure's asset-value log-return random variable r_n depends on two composite factors H_n (hierarchy composite factor) and S_n (sector composite factor) according to following formula:

$$\overline{r_n} = g_n \cdot H_n + \sqrt{1 - g_n^2} \cdot \varepsilon_n, \tag{10}$$

$$r_n = \sqrt{1 - \omega_n^2} \cdot S_n + \omega_n \cdot \overline{r_n}, \tag{11}$$

where H_n is composite factor of hierarchy correlation risk factors (HSF), $g_n \in (0, 1)$ is group correlation coefficient with composite HSF, S_n is composite factor of sector correlation risk factors (SSF), $\omega_n \in (0, 1)$ is idiosyncratic weight towards composite SSF and ε_n is exposure idiosyncratic factor.

Let K_S denote the number of SSF and K_H denote the number of HSF. We assume that, there are corresponding K_S sector composite factors and K_H hierarchy composite factors. Links (correlation) between single composite factors are represented differently for HSF and SSF.

In the case of HSF we assume links between systemic factors take form of a dependence tree structure. Let $H_{(1)}, \ldots, H_{(K_H)}$ denote the unique composite factors of HSF corresponding to K_H. Composite factors are ordered according to a given tree structure and their calculation is given recursively, where every node $H_{(k)}$ has at most one parent $H_{(l)}$ and specified correlation coefficient g_k^H, see formula (12).

$$H_{(k)} = \begin{cases} g_k^H H_{(l)} + \sqrt{1 - (g_k^H)^2} \varepsilon_k^H, & p(k) = l \\ \varepsilon_k^H, & p(k) = \emptyset \end{cases}, \tag{12}$$

where ε_k^H denotes idiosyncratic term for HSF k and $p(k)$ is parent mapping function. Example of calculating HSF composite factors can be seen in Figure 1.

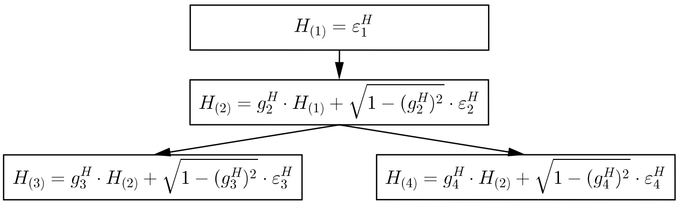
$H_{(1)} = \varepsilon_1^H$
$H_{(2)} = g_2^H \cdot H_{(1)} + \sqrt{1 - (g_2^H)^2} \cdot \varepsilon_2^H$
$H_{(3)} = g_3^H \cdot H_{(2)} + \sqrt{1 - (g_3^H)^2} \cdot \varepsilon_3^H$
$H_{(4)} = g_4^H \cdot H_{(2)} + \sqrt{1 - (g_4^H)^2} \cdot \varepsilon_4^H$

Figure 1: Example of group correlation tree

In the case of SSF we assume links between systemic factors take form of the full correlation matrix. Let $S_{(1)}, \ldots, S_{(K_S)}$ denote unique composite factors of SSF. Single composite factors $S_{(1)}, \ldots, S_{(K_S)}$ are defined by a given correlation matrix Σ and are calculated as

$$\begin{bmatrix} S_{(1)} \\ \vdots \\ S_{(K_S)} \end{bmatrix} = \sqrt{\Sigma} \cdot \begin{bmatrix} \varepsilon_1^S \\ \vdots \\ \varepsilon_{K_S}^S \end{bmatrix}, \tag{13}$$

where ε_k^S denote idiosyncratic term for SSF k.

All of the aforementioned parameters g_n, ω_n, g_k^H, HSF tree structure and correlation matrix Σ are given as portfolio parameters and can be interpreted in the standard form of $\alpha_{n,k}$ and β_n parameters, where $\sum_{k=1}^{K} \alpha_{n,k}^2 = 1$ is satisfied.

For tested model the LGDs are considered from the Beta distribution with mean and standard deviation given by portfolio parameters.

For a better illustration in the further text we will use normalized EAD_n :

$$\sum_{k=1}^{N} EAD_k = 1, \tag{14}$$

which express EAD_n as a portion of the total portfolio exposure.

2 EMPLOYING IMPORTANCE SAMPLING

As mentioned before, we are interested in the VaR and the ES of the observed portfolio loss random variable L_N. The Monte Carlo approximation of these values is highly sensitive to the stated *confidence level* p, which is usually very close to 1. In our study we use the confidence levels of $0.99995, 0.9995$ and 0.995. For example when the confidence level is 0.99995 the MC simulation of 10^6 samples provides only 50 samples with the information about VaR/ES.

One of the straightforward ways to increase the number of samples in the region of VaR/ES calculation is to change the distribution of the portfolio loss random variable so called the importance sampling (IS) method. The principle of the IS can be easily demonstrated on the ES calculation. The ES can be represented as the conditional mean or mean of the specific function

$$H_p(x) = \begin{cases} 0, & x < VaR_p(L_N) \\ \frac{x}{1-p}, & x \geq VaR_p(L_N) \end{cases} \tag{15}$$

$$\begin{aligned} ES_p(L_N) &= \mathbb{E}_f(H_p(L_N)) = \\ &= \int\limits_{\Omega} H_p(L_N^*(\overline{y})) \cdot f(\overline{y}) \, d\overline{y}, \end{aligned} \tag{16}$$

where $\overline{y}$ are values of the random vector $\overline{Y}$ of all random variables contributing to L_N (idiosyncratic terms, LGDs), Ω is the set of the all possible values of $\overline{y}$, $f(\overline{y})$ is the joint probability density function of $\overline{Y}$, $L_N^*(\overline{y}) : L_N^*(\overline{Y}) = L_N$ is the function mapping $\overline{y}$ to corresponding value of L_N and $\mathbb{E}_f$ is mean under the pdf $f(\overline{y})$. If we use the IS with the new probability distribution of L_N given by pdf $g(\overline{y})$ we can calculate original ES as

$$\mathbb{E}_g \left(H_p \left(L_N^*(\overline{Y}) \right) \cdot \frac{f(\overline{Y})}{g(\overline{Y})} \right) =$$

$$= \int_{\Omega} H_p\left(L_N^*(\overline{y})\right) \cdot \frac{f(\overline{y})}{g(\overline{y})} \cdot g(\overline{y})\, d\overline{y} =$$

$$= \mathbb{E}_f(H_p(L_N)) = ES_p(L_N). \tag{17}$$

The ratio of probability density functions $w(\overline{y}) := \frac{f(\overline{y})}{g(\overline{y})}$ is called the *the likelihood ratio* (LR). From formula (17) we can see the natural requirement on $g(\overline{y}) : H_p\left(L_N^*(\overline{y})\right) \cdot f(\overline{y}) > 0 \Rightarrow g(\overline{y}) > 0$. Formula (17) also provide the MC estimation of the ES when using the IS

$$\overline{ES_p^g}(L_N) \;=\; \frac{1}{N}\sum_{i=1}^{M} H_p\left(L_N^*\left(\overline{Y_i}\right)\right) \cdot w(\overline{Y_i}) =$$

$$= \frac{\sum_{i=1}^{M} L_N^*(\overline{Y_i})\left(L_N^*(\overline{Y_i}) \geq \overline{VaR_p^g}(L_N)\right) w(\overline{Y_i})}{M \cdot (1-p)}, \tag{18}$$

where $\overline{Y_i}$ is i-th sample of $\overline{Y} \sim g(\overline{y})$ and M is the number of random samples. It remains to define $\overline{VaR_p^g}(L_N)$ as

$$\overline{VaR_p^g}(L_N) = \min\left\{L_N^*(\overline{Y_i}) : \psi\left(\overline{Y_i}\right) \leq (1-p)\cdot M\right\}, \tag{19}$$

where $\psi\left(\overline{Y_i}\right) := \sum_{j=1}^{M}\left(L_N(\overline{Y_i}) > L_N^*(\overline{Y_j})\right) \cdot w(\overline{Y_j})$.

2.1 *Cross-Entropy method*

We already know the principles of the IS and have the IS estimators of VaR and ES, but a new IS pdf $g(\overline{y})$ is still unknown. The most straightforward method for the estimation of $g(\overline{y})$ is to minimize the variance of the ES IS estimator:

$$g(\overline{y}) = \underset{v(\overline{y})\in\mathcal{X}}{\arg\min}\left\{S_v^2\left(H\left(L_N^*\left(\overline{Y}\right)\right)\frac{f\left(\overline{Y}\right)}{v\left(\overline{Y}\right)}\right)\right\}, \tag{20}$$

where $S_v^2\,(X)$ denote variance according to pdf $v\,(\overline{y})$ and $\mathcal{X}$ is an arbitrary system of the pdfs fulfilling the condition $v(\overline{y}) : H_p\left(L_N^*(\overline{y})\right) \cdot f(\overline{y}) > 0 \Rightarrow v(\overline{y}) > 0$. This approach is called *variance minimization* (VM) method. Usually the VM method leads to very difficult problems, which have to be solved numerically. Another approach to obtain the IS pdf $g(\overline{y})$ is the cross-entropy (CE) method. The CE method similarly to the VM method solve a minimization problem, but instead of minimizing the variance it minimize the Kullback-Leibler (KL) divergence $D(g^*, v)$ with the optimal (zero variance) IS distribution

$$g^*\,(\overline{y}) = \frac{\left|H_p\left(L_N^*\,(\overline{y})\right)\right| \cdot f\,(\overline{y})}{\mathbb{E}_f\left(\left|H_p\left(L_N\right)\right|\right)} : \tag{21}$$

$$g(\overline{y}) \;:=\; \underset{v(\overline{y})\in\mathcal{X}}{\arg\min}\left\{D(g^*(\overline{y}), v(\overline{y}))\right\} =$$

$$= \underset{v(\overline{y})\in\mathcal{X}}{\arg\min}\left\{\int_{\Omega} g^*(\overline{y}) \ln\frac{g^*(\overline{y})}{v(\overline{y})} d\overline{y}\right\} =$$

$$= \underset{v(\overline{y})\in\mathcal{X}}{\arg\max}\left\{\int_{\Omega} |H_p(L_N^*(\overline{y}))| f(\overline{y}) \ln v(\overline{y}) d\overline{y}\right\}. \tag{22}$$

To obtain a solvable problem, we need to add some constrain to the system of pdfs $\mathcal{X}$. Usual choice is a parametrized family of pdfs:

$$\mathcal{X} := \left\{v\left(\overline{x};\theta\right) \forall\theta \in \Theta\right\}, \tag{23}$$

where $v\left(\overline{x};\theta\right)$ is pdf taking vector of parameters θ and $\Theta := \left\{\theta : H_p\left(L_N^*(\overline{y})\right) \cdot f(\overline{y}) > 0 \Rightarrow v(\overline{y};\theta) > 0\right\}$. Obtained minimization problem is usually concave, therefore we can replace the optimization problem with the following equation

$$\theta : \int_{\Omega} |H_p(L_N^*(\overline{y}))| f(\overline{y}) \nabla_\theta \ln v(\overline{y};\theta) d\overline{y} = 0. \tag{24}$$

To solve the problem (24) we use the Monte Carlo simulation:

$$\theta : \sum_{i=1}^{M} |H_p(L_N^*(\overline{Y_i}))| \nabla_\theta \ln v(\overline{Y_i};\theta) = 0, \tag{25}$$

this is called the *stochastic counterpart* (SC) of the problem (24). Note that (25) is usually a system of non-linear equations, but for some pdfs results into an explicit solution.

In this paper we focus mainly on the IS of idiosyncratic terms of the systemic factors (HSF and SSF). Therefore to simplify the notation of the random vector $\overline{Y}$ of all random variables contributing to L_N will be in further text understood as a vector of $K_S + K_H$ independent standard normal random variables. LGDs or other random variables will be still part of $\overline{Y}$, but the IS won't affect them.

Now if we consider $\mathcal{X}$ as a system of $K_S + K_H$ independent normally distributed random variables parametrized by mean and variance, we will get the following solution of problem (25):

$$\widetilde{\mu_j} = \frac{\sum_{i=1}^{M} |H_p(L_N^*(\overline{Y_i}))| \left(\overline{Y_i}\right)_j}{\sum_{i=1}^{M} |H_p(L_N^*(\overline{Y_i}))|}, \forall j, \tag{26}$$

$$\widetilde{\sigma_j}^2 = \frac{\sum_{i=1}^{M} |H_p(L_N^*(\overline{Y_i}))| \left(\left(\overline{Y_i}\right)_j - \widetilde{\mu_j}\right)^2}{\sum_{i=1}^{M} |H_p(L_N^*(\overline{Y_i}))|}, \forall j, \tag{27}$$

where $\widetilde{\mu_j}, \widetilde{\sigma_j}^2$ is the SC approximation of mean,variance of j-th component of $\overline{Y}$ and $\left(\overline{Y_i}\right)_j$ is j-th component of i-th MC sample.

2.2 *Gaussian mixture model*

In the end of previous part we presented formulas for calculating the "optimal" IS distribution in the family of normal distributions. This approach is commonly used for the IS in the multi-factor Merton model, see for example (Glasserman & Li 2005). The choice of the IS family of distributions as normal distributions is not always optimal and can improved by more complex IS family of distributions.

The IS family of distributions examined in this paper is the family of the Gaussian mixture distributions, the same approach in different application can be found in (Kurtz & Song 2013). The Gaussian mixture random variable is defined as a weighted sum of different normal random variables. The pdf of the Gaussian mixture random variable can be expressed as

$$g\left(x;\boldsymbol{p},\boldsymbol{\mu},\boldsymbol{\sigma}\right)=\sum_{i=1}^{n}p_i\cdot f_N\left(x;\mu_i,\sigma_i\right), \qquad (28)$$

where $f_N\left(x;\mu_i,\sigma_i\right)$ is the pdf of the normal distribution with the mean μ_i and the variance σ_i^2 and $\|\boldsymbol{p}\|_1=\sum\limits_{i=1}^{n}p_i=1$. New IS Gaussian mixture joint pdf of $\overline{Y}$ will be

$$g_{\overline{Y}}\left(\overline{x};\overline{\boldsymbol{p}},\overline{\boldsymbol{\mu}},\overline{\boldsymbol{\sigma}}\right)=\prod_{j=1}^{K_S+K_H}g\left(x_j;\boldsymbol{p}_j,\boldsymbol{\mu}_j,\boldsymbol{\sigma}_j\right), \qquad (29)$$

where $\overline{\boldsymbol{p}},\overline{\boldsymbol{\mu}},\overline{\boldsymbol{\sigma}}$ are matrices of K_S+K_H columns of parameters $\boldsymbol{p}_j,\boldsymbol{\mu}_j,\boldsymbol{\sigma}_j$. Therefore the system of pdfs for the IS is

$$\mathcal{X}:=\left\{g_{\overline{Y}}\left(\overline{x};\overline{\boldsymbol{p}},\overline{\boldsymbol{\mu}},\overline{\boldsymbol{\sigma}}\right):\left\|\boldsymbol{p}_j\right\|_1=1,\sigma_{j,i}>0\right\}. \qquad (30)$$

Because the support of the pdf of the normal distribution is $\mathbb{R}$, the condition $f(\overline{x})>0\Rightarrow g_{\overline{Y}}\left(\overline{x};\overline{\boldsymbol{p}},\overline{\boldsymbol{\mu}},\overline{\boldsymbol{\sigma}}\right)>0$ is fulfilled. Since the components of $g_{\overline{Y}}\left(\overline{x};\overline{\boldsymbol{p}},\overline{\boldsymbol{\mu}},\overline{\boldsymbol{\sigma}}\right)$ are independent, the problem (24) reduces into K_S+K_H systems of non-linear equations. Therefore together with the condition $\left\|\boldsymbol{p}_j\right\|_1=1$ we will receive $\forall j=1,\ldots,K_S+K_H,\forall i=1,\ldots,n:$

$$\mu_{j,i}=\frac{\sum\limits_{k=1}^{M}\left|H_p\left(L_N^*\left(\overline{Y_k}\right)\right)\right|\gamma_{k,j,i}\left(\overline{Y_k}\right)_j}{\sum\limits_{k=1}^{M}\left|H_p\left(L_N^*\left(\overline{Y_k}\right)\right)\right|\gamma_{k,j,i}},$$

$$\sigma_{j,i}^2=\frac{\sum\limits_{k=1}^{M}\left|H_p\left(L_N^*\left(\overline{Y_k}\right)\right)\right|\gamma_{k,j,i}\left(\left(\overline{Y_k}\right)_j-\mu_{j,i}\right)^2}{\sum\limits_{k=1}^{M}\left|H_p\left(L_N^*\left(\overline{Y_k}\right)\right)\right|\gamma_{k,j,i}},$$

$$p_{j,i}=\frac{\sum\limits_{k=1}^{M}\left|H_p\left(L_N^*\left(\overline{Y_k}\right)\right)\right|\gamma_{k,j,i}}{\sum\limits_{k=1}^{M}\left|H_p\left(L_N^*\left(\overline{Y_k}\right)\right)\right|}, \qquad (31)$$

where

$$\gamma_{k,j,i}:=\frac{p_{j,i}\cdot f_N\left(\left(\overline{Y_k}\right)_j;\mu_{j,i},\sigma_{j,i}\right)}{\sum\limits_{i=1}^{n}p_{j,i}\cdot f_N\left(\left(\overline{Y_k}\right)_j;\mu_{j,i},\sigma_{j,i}\right)}. \qquad (32)$$

We obtain K_S+K_H systems, each representing a problem of approximation of the Gaussian mixture from data sample. This sub-problems can be solved for example by EM or K-means algorithm see (Bishop 2006, Redner & Walker 1984).

But the computation effort of the system (31) will be significantly smaller if we have an information from which component of $g\left(x_j;\boldsymbol{p}_j,\boldsymbol{\mu}_j,\boldsymbol{\sigma}_j\right)$ was $\left(\overline{Y_k}\right)_j$ generated. Let $\overline{z}_{k,j}$ denote Bernoulli vector of identificators, such as

$$\left(\overline{z}_{k,j}\right)_i=\begin{cases}1, & \left(\overline{Y_k}\right)_j\sim f_N\left(x;\mu_{j,i},\sigma_{j,i}\right)\\ 0, & \text{otherwise}\end{cases}. \qquad (33)$$

One can show that if we know the values of $\overline{z}_{k,j}$, then $\gamma_{k,j,i}=\left(\overline{z}_{k,j}\right)_i$. Therefore the system (31) results in explicit solution of the problem (24).

2.3 *Objective function for component identification*

In the previous part we constructed formulas for the calculation of the IS Gaussian mixture distribution. These formulas depend on the knowledge of the sample's source component $\overline{z}_{k,j}$, but this is not easily obtainable information. In this part we propose a numerical approximation of $\overline{z}_{k,j}$ based on model behaviour.

First let's consider a set of K_S+K_H functions

$$\psi_j\left(\overline{y}\right):=\frac{\sum\limits_{i=1}^{N}EAD_i\cdot D_i\left(\overline{y}\right)\cdot\beta_i\cdot\alpha_{i,j}}{\max\limits_{i=1,\ldots,N}\left\{\beta_i\cdot\alpha_{i,j}\right\}\cdot\sum\limits_{i=1}^{N}EAD_i\cdot D_i\left(\overline{y}\right)}, \qquad (34)$$

where $\beta_i,\alpha_{i,j},EAD_i$ are portfolio parameters of exposure i and $D_i\left(\overline{y}\right)$ is the default indicator of exposure i under the vector of all idiosyncratic shocks $\overline{y}$. In the case of no defaulting exposure the function $\psi_j\left(\overline{y}\right)$ yields 0. It can be easily shown that $0\leq\psi_j\left(\overline{y}\right)\leq 1$.

To demonstrate a link between $\overline{z}_{k,j}$ and $\psi_j\left(\overline{Y_k}\right)$ let's consider portfolio containing a component j with huge impact on L_N. In Figure 2 we show dependence between component idiosyncratic shock X_j and $\psi_j\left(\overline{y}\right)$ under the condition $L_N\geq VaR_p(L_N)$. From the study of the aforementioned figure we can conclude, that:

- X_j distribution under the condition $L_N\geq VaR_p(L_N)$ consist of multiple components,
- $\psi_j\left(\overline{y}\right)$ separate these components by it's value, in other words we can assume

$$\left(\psi_j\left(\overline{Y_k}\right)\in\left(a_i,a_{i+1}\right)\right)\Rightarrow\left(\left(\overline{z}_{k,j}\right)_i=1\right), \qquad (35)$$

where $0 = a_1 \leq \ldots \leq a_{n+1} = 1$ (n denote number of the Gaussian mixture components) are some known values.

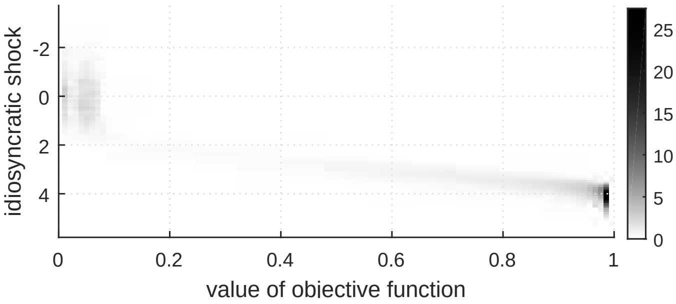

Figure 2: Approximation of dependence between $\psi_j\left(\overline{y}\right)$ and X_j

Numerical justification of the assumption (35) can be seen in Figure 3, where we can see the histogram of the simulation of X_j distribution under the condition $L_N \geq VaR_p(L_N)$ and it's approximation by the 3 component Gaussian mixture in comparison with approximation by the normal distribution. Approximation by the Gaussian mixture was obtained by using the objective function $\psi_j\left(\overline{y}\right)$ and the precalculated bounds $a_1 = 0, a_2 = 0.2, a_3 = 0.8, a_4 = 1$. Other fact beside very good approximation obtained from the proposed procedure is that the approximation obtained by the normal distribution differ significantly from the approximated distribution. Note that X_j distribution under the condition $L_N \geq VaR_p(L_N)$ is an optimal distribution found by the CE method for $H_p(x) = (x \geq VaR_p(L_N))$.

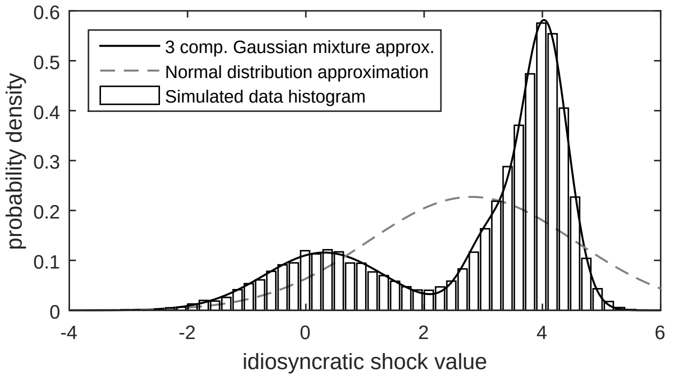

Figure 3: Approximation of X_j distribution under the condition $L_N \geq VaR_p(L_N)$ by 3 component Gaussian mixture

Since we want to calculate both VaR and ES, the CE problem formulation based on $H_p(x)$ given by (15) does not have to be optimal. The VaR approximation can suffer if the CE method favours samples with very high value of loss and disfavours those close to $VaR_p(L_N)$ bound. Therefore we will use

$$H_p(x) = \left(x \geq VaR_p(L_N)\right), \tag{36}$$

which give all samples with $L_N \geq VaR_p(L_N)$ same weight.

Till now we haven't dealt with bounds a_i calculation. Generally it can be a difficult problem, but $\psi_j\left(\overline{y}\right)$ component recognition is not sensitive to small changes of a_i, therefore rough approximation is sufficient. Such computationally feasible sufficient approximation can be obtained by minimizing (by e.q. line-search methods) difference between MC sample of X_j distribution under the condition $L_N \geq VaR_p(L_N)$ and the Gaussian mixture obtained using $\psi_j\left(\overline{y}\right)$ component recognition.

2.4 *Adaptive CE method for IS calculation*

So far we have constructed formulas for calculating the Gaussian mixture IS, stated the optimal form of the function $H_p(x)$ in (36) and constructed an instrument for the Gaussian mixture j-th component identification using objective function $\psi_j\left(\overline{y}\right)$. But the single calculation from M MC samples would result in the poor approximation, if the M was not high enough. The sufficient number of the MC samples for stable and precise approximation of the CE problem is comparable with the number of MC samples for sufficient approximation of VaR/ES. This would make the whole IS principle useless, because it won't bring savings in the computational time/effort. Solution to this inconvenience is iterative process, slowly shifting the IS distribution to the CE method optimal one.

The formulas for the CE method SC (31) can be modified by using the IS during the SC process:

$$\widetilde{\mu}_{j,i,t} = \frac{\sum_{k=1}^{M} \left|H_p\left(\overline{Y_k}\right)\right| w\left(\overline{Y_k}\right) \left(\overline{z}_{k,j}\right)_i \left(\overline{Y_k}\right)_j}{\sum_{k=1}^{M} \left|H_p\left(\overline{Y_k}\right)\right| w\left(\overline{Y_k}\right) \left(\overline{z}_{k,j}\right)_i},$$

$$\widetilde{\sigma}^2_{j,i,t} = \frac{\sum_{k=1}^{M} \left|H_p\left(\overline{Y_k}\right)\right| w\left(\overline{Y_k}\right) \left(\overline{z}_{k,j}\right)_i \left(\left(\overline{Y_k}\right)_j - \widetilde{\mu}_{j,i,t}\right)^2}{\sum_{k=1}^{M} \left|H_p\left(\overline{Y_k}\right)\right| w\left(\overline{Y_k}\right) \left(\overline{z}_{k,j}\right)_i},$$

$$\widetilde{p}_{j,i,t} = \frac{\sum_{k=1}^{M} \left|H_p\left(\overline{Y_k}\right)\right| w\left(\overline{Y_k}\right) \left(\overline{z}_{k,j}\right)_i}{\sum_{k=1}^{M} \left|H_p\left(\overline{Y_k}\right)\right| w\left(\overline{Y_k}\right)}, \tag{37}$$

where t denotes iteration, $\left(\overline{z}_{k,j}\right)_i$ denote if the i-th component of j-th systemic factor's Gaussian mixture was the source of the sample k, $H_p\left(\overline{Y_k}\right) := \left(L_N^*\left(\overline{Y_k}\right) \geq VaR_p(L_N)\right)$ and

$$w\left(\overline{Y_k}\right) = \frac{f\left(\overline{Y_k}\right)}{g_{\overline{Y}}\left(\overline{Y_k}; \overline{\boldsymbol{p}}_{t-1}, \overline{\boldsymbol{\mu}}_{t-1}, \overline{\boldsymbol{\sigma}}_{t-1}\right)}, \tag{38}$$

where $f\left(\overline{Y_k}\right)$ is the pdf of nominal distribution (joint distribution of the independent normal distributions) and $g_{\overline{Y}}\left(\overline{Y_k}; \overline{\boldsymbol{p}}_{t-1}, \overline{\boldsymbol{\mu}}_{t-1}, \overline{\boldsymbol{\sigma}}_{t-1}\right)$ is the pdf of IS Gaussian mixture distribution given by parameters approximated in the iteration $t-1$.

In the definition of $H_p\left(\overline{Y_k}\right)$ is still present the unknown value of $VaR_p(L_N)$, which can be replaced by it's approximation $\overline{VaR_p^{g_Y}}(L_N)$ from t-th iteration. The last obstacle is that the $H_p\left(\overline{Y_k}\right)$ will be for most samples zero and the iteration process will crash at the beginning. The solution to this is the replacement of the confidence level p by a sequence of p_i which is at first few iteration significantly lower than p and at the and of iterative process equals p.

All of the previous observations lead to algorithm 1. Obtained algorithm can be further enhanced for example by the Screening method or by the adaptive smoothing parameter sequence see (Kroese, Taimre, & Botev 2013, Rubinstein & Kroese 2013, Rubinstein & Kroese 2011).

Algorithm 1 Adaptive iterative calculation of the CE problem

```
Inputs: $\overline{\boldsymbol{p}}_0, \overline{\boldsymbol{\mu}}_0, \overline{\boldsymbol{\sigma}}_0$, for every systemic factor $j$ sequence of bounds $a_1 \leq \ldots \leq a_{n+1}$, smoothing parameter $\alpha \in (0,1)$, sequence of $p_1, p_2, \ldots, p_i, p, p, \ldots$, sequence of sample sizes $M_t$, set $t = 1$
  1. Simulate $M_t$ samples $\overline{Y_1}, \ldots, \overline{Y_{M_t}}$ from the Gaussian mixture distribution given by parameters $\overline{\boldsymbol{p}}_{t-1}, \overline{\boldsymbol{\mu}}_{t-1}, \overline{\boldsymbol{\sigma}}_{t-1}$, calculate $\overline{VaR_{pt}^{g_Y}}(L_N)$, $H_{p_t}\left(\overline{Y_k}\right), w\left(\overline{Y_k}\right), \overline{z}_{k,j}\, \forall j$.
  2. Calculate $\widetilde{\boldsymbol{p}}_{t-1}, \widetilde{\boldsymbol{\mu}}_{t-1}, \widetilde{\boldsymbol{\sigma}}_{t-1}$ using formula (37).
  3. Update parameters:
     $\overline{\boldsymbol{p}}_t = \alpha \cdot \widetilde{\boldsymbol{p}}_{t-1} + (1-\alpha) \cdot \overline{\boldsymbol{p}}_{t-1}$,
     $\overline{\boldsymbol{\mu}}_t = \alpha \cdot \widetilde{\boldsymbol{\mu}}_{t-1} + (1-\alpha) \cdot \overline{\boldsymbol{\mu}}_{t-1}$,
     $\overline{\boldsymbol{\sigma}}_t = \alpha \cdot \widetilde{\boldsymbol{\sigma}}_{t-1} + (1-\alpha) \cdot \overline{\boldsymbol{\sigma}}_{t-1}$
  4. If some stopping condition is fulfilled (e.g. $\overline{\boldsymbol{p}}_t, \overline{\boldsymbol{\mu}}_t, \overline{\boldsymbol{\sigma}}_t \approx \overline{\boldsymbol{p}}_{t-1}, \overline{\boldsymbol{\mu}}_{t-1}, \overline{\boldsymbol{\sigma}}_{t-1}$) return the approximation of optimal parameters $\overline{\boldsymbol{p}}_t, \overline{\boldsymbol{\mu}}_t, \overline{\boldsymbol{\sigma}}_t$, if not set $t = t + 1$ and go back to step 1.
Note: sequences $p_t$ and $M_t$ should be calculated inside the iterative process with respect to current sample $\overline{Y_1}, \ldots, \overline{Y_{M_t}}$ (e.g. from the position of sample representing $\overline{VaR_{pt}^{g_Y}}(L_N)$ in sorted sequence $L_N^*\left(\overline{Y_k}\right)$)
```

3 IMPLEMENTATION AND GPU PARALLELIZATION

The serial Matlab implementation is a straightforward interpretation of the multi-factor Merton model with the Matlab built-in functions. The whole simulation (all of the MC samples) can be calculated at once without the use of loops. Most computationally expensive parts of the simulation can be calculated by very well optimized Matlab matrix functions and therefore this implementation can serve as a good comparison tool of the performance efficiency for further GPU implementations.

3.1 *GPU parallelization*

As was already mentioned the simulation of the multifactor Merton model consists of many MC samples, that are mutually independent. This is suitable for a massively parallel computation hardware such as the GPU device.

3.1.1 *Shortly about GPUs*

Let us very shortly outline main parameters of GPUs, which are crucial for model implementation:

- GPUs consist of many (in current devices in order of thousands) computation cores, grouped into *streaming multiprocessors* (SM), communication between single cores is strictly restricted to groups belonging to one SM unit. Execution of CUDA kernel (parallel GPU implementation) must mirror this structure and we must specify block size (how many threads per SM will run) and grid size (how many blocks will be executed).
- There are four basic types of memory on the GPUs:
  - global memory: main storage memory, large, high latency (thread waits long time before get the data), must be accessed in pattern (i-th core access i-th element) to obtain reasonable utilization of bandwidth
  - shared memory: small, shared between cores in one SM, low latency
  - constant memory: small, can broadcast content of array among all cores
  - registers: cannot be directly accessed, separated for every core, very fast, buffer some small local variables

For software implementation on GPU we use the NVIDIA CUDA technology. For further informations see (NVIDIA 2015).

3.1.2 *GPU implementations overview*

When implementing multi-factor Merton model we decided to create multiple implementations, which can benefit from different type of portfolios:

- "base" GPU implementation: straightforward interpretation of the model, single threads perform single MC samples in the same way as the serial implementation,
- "sparse" GPU implementation: similar to "base" implementation, but the matrix of $\alpha_{i,j}$ coefficients is handled in sparse format (only column/row index and value of non-zero elements is stored)

- “specialized” GPU implementation: is applicable only on specialized type of portfolios which use systemic factor grouping into SSF and HSF, implementation fits the mathematical description in subsection 1.2 (correlation matrix of SSF is stored in constant memory).

Finally some remarks shared by all GPU implementations:

- usage of shared memory buffering - as all cores need the same portfolio data, we can (by selected cores) copy the data from global to shared memory (which is much faster than global),
- generating random numbers from normal or uniform distribution is done by cuRAND library,
- compiled with `-use_fast_math` tag, which decreases precision of math functions in favour of speed
- Beta random number generator is not present in the cuRAND library, therefore we implemented own procedure based on rejection-sampling method see (Dubi 2000, Kroese, Taimre, & Botev 2013).

4 NUMERICAL RESULTS

In this section we test all of the aforementioned procedures and implementations. First we examine the behaviour of the GPU implementations and then we look at the variance reduction achievable by the proposed Gaussian mixture IS.

4.1 *GPU acceleration*

As was mentioned before we implemented three different approaches to simulate the multi-factor Merton model. Now we test their behaviour in comparison with the Matlab serial implementation on three different scenarios.

1. increasing number of the systemic factors which impacts majority of exposures (SSF), majority of corresponding $\alpha_{i,j}$ are non-zero
2. increasing number of systemic factors which impacts a small fraction of exposures (HSF), majority of corresponding $\alpha_{i,j}$ are zero
3. increasing number of exposures

All tests were performed on Intel Sandy Bridge E5-2470 processor (294.4 Gflops, 38.4 GB/s) and NVIDIA Kepler K20 accelerator (3520 Gflops, 208 GB/s), the serial Matlab implementation uses double precision and the GPU implementations use single precision. The theoretical performance benefit of GPU implementations is $192\times$(single core + double precision vs. all GPU cores + single precision) and the theoretical memory bandwidth benefit of the GPU implementations is $11\times$(double vs. single precision).

4.1.1 *Increasing number of SSF*

This test is designed to test implementation’s behaviour when the number of systemic factors increases while matrix of $\alpha_{i,j}$ coefficients becomes more dense. We use the sequence of portfolios with 1000 exposures, 100 HSF and the sequence of $(16, 25, 36, 49, 64, 81, 100)$ SSF. The density of matrix of $\alpha_{i,j}$ coefficients rises from 16% up to 51%. The scaling results can be seen in Figure 4.

From results we can observe following

- “specialized” GPU implementation’s speed-up drops from factor $515\times$(for 16 SSF) to factor $209\times$(for 100 SSF),
- “sparse” GPU implementation suffers the most, the speed-up drops from factor $77\times$(for 16 SSF.) to factor $16\times$(for 100 SSF), this could be expected because size of sparse interpretation equals $3\times$ number of non-zero elements.
- “base” GPU implementation speed-up drops from factor $35\times$(for 16 SSF) to factor $19\times$(for 100 SSF).

The drop in performance of all the GPU implementations is caused by the increasing memory complexity, which bounds the computation utilization.

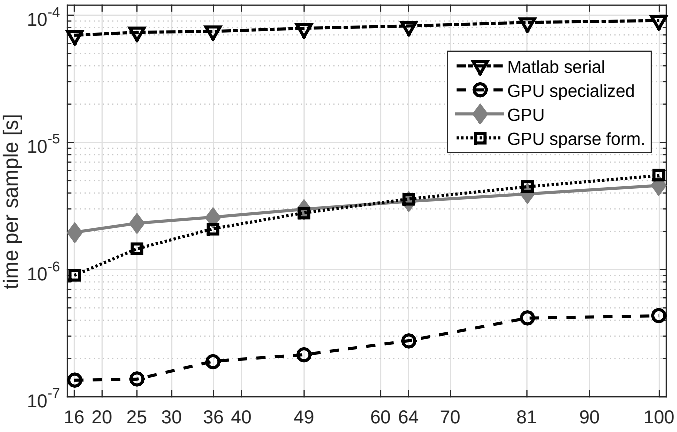

Figure 4: Implementations scaling based on rising number of high impact systemic factors

4.1.2 *Increasing number of HSF*

The second test is designed as the counter example to the first one. Now we test the sequence of portfolios with 1000 exposures, 25 SSF and sequence of $(100, 200, 400, 800, 1600)$ HSF. The density of matrix of $\alpha_{i,j}$ coefficients decreases from 22% down to 1.7%. The results can be seen in Figure 5.

From results we can observe following

- “specialized” GPU implementation speed-up rise from factor $537\times$(for 100 HSF) to factor $1001\times$(for 1600 HSF),
- “sparse” GPU implementation benefits the most, speed-up rise from factor $51\times$(for 100 HSF) to factor $287\times$(for 1600 HSF), this could be again expected because number of non-zero elements of matrix of $\alpha_{i,j}$ coefficients does not increase much.
- “base” GPU implementation speed-up drops from factor $32\times$(for 100 HSF) to factor $18\times$(for 1600 HSF).

The drop in performance of “base” GPU implementation is caused again by the increasing memory complexity, because it does not take in account the sparsity of matrix of $\alpha_{i,j}$ coefficients.

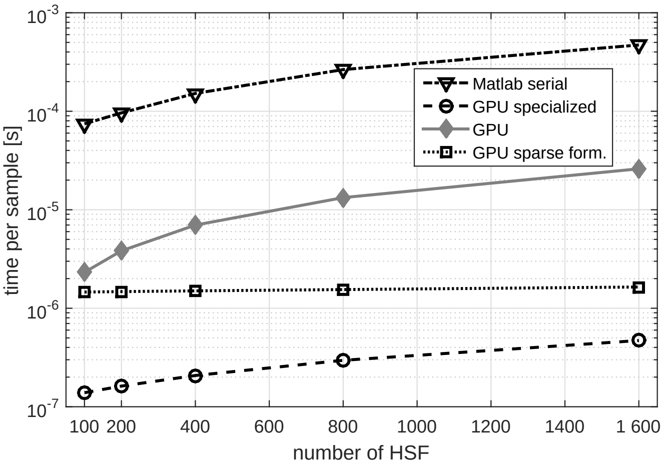

Figure 5: Implementations scaling based on rising number of low impact systemic factors

4.1.3 *Increasing number of exposures*

The last test serves as insight of the implementations behaviour when applied on the very large portfolios. We test the sequence of portfolios with 25 SSF, 100 HSF and sequence of $(1000, 2000, 4000, 8000, 16000, 32000)$ exposures. The results can be seen in Figure 6.

From results we can observe following

- “specialized” GPU implementation speed-up rise from factor $537\times$(for 100 exposures) to factor $784\times$(for 3200 exposures),
- “sparse” GPU implementation speed-up is approximately $50\times$ for all tested portfolios,
- “base” GPU implementation speed-up is approximately $30\times$ for all tested portfolios.

All of the GPU implementations exhibit good scaling when the number of exposures rises, even more the “specialized” GPU implementation benefits from the large portfolios.

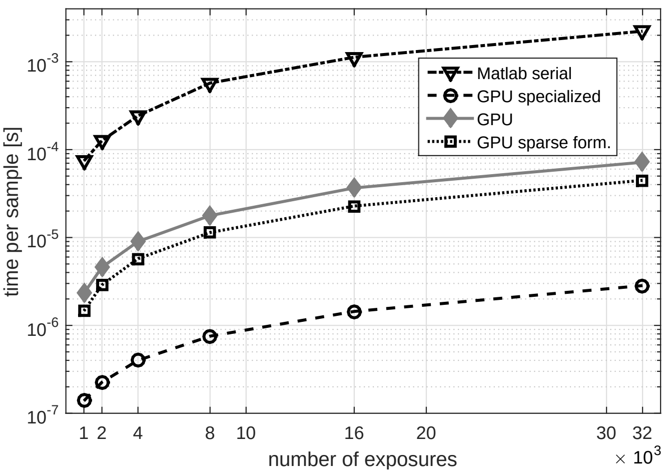

Figure 6: Implementations scaling based on rising number of exposures

4.2 *IS variance reduction*

In this part we examine the variance reduction achievable by the IS. We compare the standard IS approach using the family of normal distributions and the IS with the Gaussian mixture family of distributions.

4.2.1 *Portfolio parameters specification*

For numerical tests we constructed four different portfolios according to the structure mentioned in section 1.2. Each of the constructed portfolios consists of $N = 10^4$ exposures, $K_S = 25$ SSF and $K_H = 600$ HSF. Properties which are shared by all of the constructed portfolios are

- $EAD_i = i^2 / \sum_{j=1}^{N} j^2$,
- $PD_i = 0.001 + 0.001 \cdot \left(1 - \frac{i}{N}\right)$,
- the distribution of LGDs is Beta distribution with mean $ELGD_n = 0.5$ and standard deviation $VLGD_n = 0.25$ for all exposures,
- the structure of HSF correlation is defined by the tree template shown in Figure 7. duplicated 60 times, correlation coefficients $g_k^H = 0.9, \forall k = 1, \ldots, K_H$.
- the SSF correlation matrix is defined by 5 region and 5 industry factors, each SSF represent unique combination of the region and the industry. Correlation between two SSF is 0.2 if they share same region, 0.15 if they share same industry and 0.03 otherwise.
- exposures are assigned to a composite SSF/HSF randomly by defined probabilities $p_k^S = \mathbb{P}\left(S_n = S_{(k)}\right)$ and $p_k^H = \mathbb{P}\left(H_n = H_{(k)}\right)$.

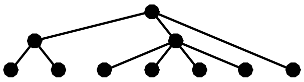

Figure 7: Template structure of HSF correlation tree

Single portfolios differs in exposure assignation to SSF, HSF and coefficients g_n, ω_n.

Portfolio 1. $p_k^S \sim \ln N(0, 0.5)$ and normalized, $p_k^H \sim \ln N(0, 10)$ and normalized, $g_n = 0.9, \omega_n = 0.5$.

Portfolio 2. $p_k^S = \frac{1}{K_S}, p_k^H = \frac{1}{K_H}, g_n = 0.9, \omega_n = 0.5$

Portfolio 3. $p_k^S \sim \ln N(0, 0.5)$ and normalized, $p_k^H \sim \ln N(0, 10)$ and normalized, $g_n = 0.5, \omega_n = 0.9$

Portfolio 4. $p_k^S = \frac{1}{K_S}, p_k^H = \frac{1}{K_H}, g_n = 0.5, \omega_n = 0.9$

Portfolio 1. represents a portfolio with clustered exposures (large groups of exposures with the same HSF/SSF composite factor) with high dependence on the systemic factors.

Portfolio 2. has the same level of exposure dependence on the systemic factors as portfolio 1., but exposures are equally distributed among the HSF/SSF composite factors.

Portfolio 3. has exposures clustered as in portfolio 1., but the level of exposure dependence is as low as in portfolio 2.

Portfolio 4. has exposures evenly distributed as portfolio 2. and low level of exposure dependence as in portfolio 3.

4.2.2 *Variance reduction in comparison with the standard approach*

Beside different portfolios we also test different levels of *confidence level* $p \in \{0.99995, 0.9995, 0.995\}$. First lets examine VaR and ES of selected portfolios and *confidence levels*, VaR/ES calculated by MC using 10^7 samples are listed in Table 1.

Table 1: Tested portfolios VaR and ES

		confidence level p		
Characteristic	Portf. idx.	0.99995	0.9995	0.995
VaR	1	0.0371	0.0251	0.0129
	2	0.0291	0.0203	0.0123
	3	0.0057	0.0041	0.0027
	4	0.0051	0.0038	0.0026
ES	1	0.0417	0.0304	0.0181
	2	0.0332	0.024	0.016
	3	0.0065	0.0048	0.0033
	4	0.0058	0.0044	0.0032

Measured levels of VaR, ES shows that the lower level of exposure dependence and even distribution of exposures leads to the lower value of VaR,ES. This can suggest, that the IS for portfolio 3. and 4. could be less effective. The impact of *confidence level* is predictable, the IS effectiveness will be lower for lower *confidence levels*. This is caused by reducing rarity of samples providing information about VaR, ES and therefore no large change of the distribution is needed.

Let's proceed to the testing of the variance reduction. In Table 2 we can see the variance of all combinations of tested confidence levels and portfolios for the plain (crude) MC simulation, the IS using the normal distribution and the IS using the Gaussian mixture. The variance is calculated as an empirical value of 1000 simulations consisted of 10^6 samples.

For more illustrative view of achieved variance reduction see Figure 8. Figure shows a comparison of the variance reduction between the standard and the Gaussian mixture approach for all confidence levels and portfolios combinations. Clearly the IS using the Gaussian mixture achieve better variance reduction in every test, this was evident because the normal distributions family is a subset of the Gaussian mixture distributions family.

For exact comparison of the two IS approaches, see Table 3. Table shows ratios of the variance reduction between the IS using the normal distribution and the IS using the Gaussian mixture.

Table 3: Variance reduction ratio Gaussian mix./normal dist.

		confidence level p		
Characteristic	Portf. idx.	0.99995	0.9995	0.995
VaR	1	9.54	4.90	3.65
	2	9.10	6.35	4.26
	3	3.06	1.91	1.28
	4	3.58	2.38	1.35
ES	1	8.37	3.16	2.30
	2	7.59	6.34	4.20
	3	3.35	1.88	1.28
	4	4.11	2.54	1.52

The improvement of the IS by using the Gaussian mixture is given by the presence of systemic factor with very high impact on loss L_N. These components can be found mostly in the portfolio 1. and 2., therefore in these portfolios we obtain the best improvements in the variance reduction. Sample of such component was presented in Figure 3.

5 CONCLUSION

The objective of this paper was to speed-up the multifactor Merton model MC simulation. This was fully accomplished by the GPU implementation and the IS application.

We presented three different GPU implementations, each better for different purpose. Two of the GPU implementations solve the general multi-factor Merton model with speed-up against serial model in range of $19\times$ to $287\times$ depending on structure of portfolio, see section 4.1. Third GPU implementation was specialized, taking input in form of structure described in section 1.2. This implementation achieves speed-up in range of $209\times$ to $1001\times$ depending on the portfolio structure.

For the IS we proposed a new approach using the Gaussian mixture distribution. Using this approach we achieved a significant variance reduction improvement for the certain portfolio structures, see section 4.2.2. In comparison to the standard IS approach we got from $9.5\times$ to $1.3\times$ better results. The total

Table 2: Measured variance of Crude MC, IS normal dist. and IS 3 comp. Gaussian mixture (10^6 samples, 1000 simmulations)

		Crude Monte Carlo			IS normal distribution			IS Gaussian mixture		
		confidence level p			confidence level p			confidence level p		
Char.	Portf. idx.	0.99995	0.9995	0.995	0.99995	0.9995	0.995	0.99995	0.9995	0.995
VaR	1	4.95e-07	6.21e-08	4.88e-09	6.40e-09	2.42e-09	6.96e-10	6.71e-10	4.93e-10	1.90e-10
	2	3.41e-07	2.10e-08	2.95e-09	4.98e-09	1.01e-09	6.39e-10	5.47e-10	1.59e-10	1.50e-10
	3	1.14e-08	7.63e-10	5.73e-11	8.62e-11	2.33e-11	6.14e-12	2.81e-11	1.21e-11	4.79e-12
	4	7.34e-09	6.02e-10	4.64e-11	8.31e-11	2.23e-11	5.77e-12	2.31e-11	9.36e-12	4.25e-12
ES	1	8.64e-07	1.02e-07	1.05e-08	4.24e-09	1.17e-09	4.65e-10	5.06e-10	3.69e-10	2.02e-10
	2	6.99e-07	6.03e-08	5.25e-09	2.78e-09	9.70e-10	3.37e-10	3.66e-10	1.53e-10	8.00e-11
	3	2.81e-08	1.89e-09	1.42e-10	6.88e-11	1.90e-11	5.17e-12	2.05e-11	1.00e-11	4.01e-12
	4	1.61e-08	1.37e-09	1.13e-10	7.12e-11	1.99e-11	4.52e-12	1.73e-11	7.84e-12	2.96e-12

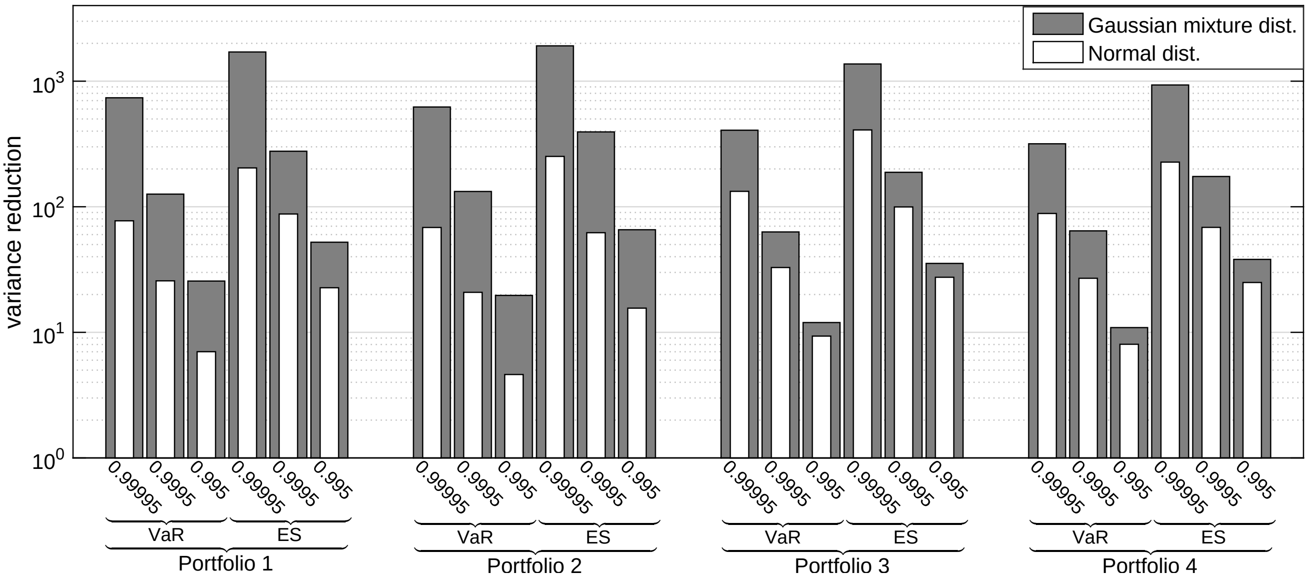

Figure 8: Variance reduction achieved by IS: Gaussian mixture and normal distribution

achieved variance reduction was up to 1911× for the ES calculation and up to 737× for the VaR calculation.

The combination of the IS and the GPU implementation can bring a speed-up of the standard serial MC simulation in orders of hundreds of thousands for portfolios with high dependence on systemic factors.

ACKNOWLEDGEMENT

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project „IT4Innovations excellence in science - LQ1602“.

REFERENCES

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Dubi, A. (2000). *Monte Carlo applications in systems engineering*. Wiley.

Glasserman, P. & J. Li (2005). Importance sampling for portfolio credit risk. *Management science 51*(11), 1643–1656.

Kroese, D. P., T. Taimre, & Z. I. Botev (2013). *Handbook of Monte Carlo Methods*. John Wiley & Sons.

Kurtz, N. & J. Song (2013). Cross-entropy-based adaptive importance sampling using gaussian mixture. *Structural Safety 42*, 35–44.

Lütkebohmert, E. (2008). *Concentration risk in credit portfolios*. Springer Science & Business Media.

NVIDIA (2015). Cuda c best practices guide. http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/. Version 7.5.

Redner, R. A. & H. F. Walker (1984). Mixture densities, maximum likelihood and the em algorithm. *SIAM review 26*(2), 195–239.

Rubinstein, R. Y. & D. P. Kroese (2011). *Simulation and the Monte Carlo method*. John Wiley & Sons.

Rubinstein, R. Y. & D. P. Kroese (2013). *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media.