UDK 519.8; 368; 65.0+004.421.2

NORKIN B.V.

ON PERFORMING ACTUARIAL CALCULATIONS ON GPU

Abstract. The paper considers issues of performing actuarial calculations on graphical processing units (GPU). Ruin probability estimation for an insurance company can always be made by means of Monte-Carlo method. In some cases it is the only applicable method. Smallness of ruin probability may require astronomical number of Monte-Carlo simulations to achieve an acceptable accuracy. Effective parallelization of the Monte Carlo method and utilizing calculation power of modern GPU makes it possible to obtain good accuracy within reasonable real time limits. A software system RMS 0.2 utilizing NVIDIA CUDA 4.2 architecture for modeling reserve evolution and performance of an insurance company was developed. Results of numerical experiments by means of this system are presented.

Introduction[1]

The essence of the insurance business is to obtain maximum net income keeping adequate insurance reserves to cover insurance claims. It is important to maintain a delicate balance between profitability and risk. The profitability is assessed on the basis of the distribution of collected dividends and of capital balance at the end of a planning horizon. Risk is measured by the probability of ruin (or insolvency) of the company.

For a formal description of insurance business classical stochastic risk process (Cramer-Lundberg model) is often used [1], which simulates a stochastic evolution of the capital of an insurance company. In this model, on one hand the capital monotonically linearly increases with time due to continuous premium incoming, and on the other hand, at random moments of time (arrivals of claims) it jumps down by a random value (size of a claim). The company is bankrupt, if the capital becomes less than zero. Obviously, this process does not reflect many aspects of the insurance company activity, for example, reinsurance, investments, dividend payments, etc. That's why, to simulate performance of a real insurance company, more realistic dynamic financial analysis models are used [2], which take into account the impact of many control and random factors on the dynamics of reserves of the company. As control parameters deductions to insurance reserves, running costs, investments, dividends, reinsurance coverage and other strategic managerial options are considered. Random factors include insurance claims, investment returns, inflation level, reimbursements due to reinsurance contracts, etc. Simulation results for fixed values of control parameters are not unique, they are random variables. A single simulation experiment does not give a complete picture of the quality of governance. One needs to simulate a large number of paths to get an idea about the probabilistic distribution of results. Having distribution, one can calculate any deterministic characteristic of the selected control strategy: the probability of ruin, expected dividends, the expected value of the residual provision, etc. However, the problem is that for the evaluation of indicators, such as the probability of ruin, one may need an astronomical number of simulations that cannot be performed in a reasonable amount of time even on modern computers. This problem was recognized in the actuarial mathematics, so that, for example, for the classical risk process a variety of approximate formulas for the ruin probability were developed [1]. But these formulas do not work for more general and more realistic models of insurance business. A universal, and often the only way to model complex dynamic stochastic systems, is the (Monte Carlo) method of statistical simulations. This method may require an astronomical number of simulations to achieve a sufficient accuracy, since the probability of ruin is usually rather small. However, parallelization makes it possible to solve this problem by increasing the processing power. In [4 - 7], a parallel version of the Monte Carlo method, along with a parallel method of successive approximations, implemented on a cluster of multiple personal computers, were used to find probability of ruin as a function of the initial capital and other parameters of an insurance company. In [8] similar issues concerning parallel random walk simulations by means of graphical accelerators with architecture NVIDIA CUDA [9] are discussed. In this paper, the Monte Carlo method for the solution of actuarial tasks is implemented on graphical accelerators with architecture NVIDIA CUDA 4.2 [9], whereby computation time was reduced by one to two orders of magnitude com-

---

[1] This article is an extended version of the report [1] at the International Conference "High Performance Computing» HPC-UA 2012, October 8-9, 2012, Kiev, Ukraine.

pared with calculations on CPU. Furthermore, the computational speed makes it possible to study dependences of the probability of ruin and of other indicators not only on the initial capital, but on a few other policy parameters of the company.

A generalized risk process as a model of an insurance company

An insurance company is obliged to maintain a certain level of insurance reserves to cover current accidental insurance claims. A mathematical model of stochastic evolution of reserves x_t of an insurance company is as follows:

$$x_t = u + \int_0^t \big(c - d(x_s)\big)\,ds - S_t,\quad 0 \le t \le T, \qquad (1)$$

where t is time variable $\left(0 \le t \le T\right)$; $x_0 = u \ge 0$ – an initial capital (insurance reserve); $S_t = \sum_{k=1}^{N_t} z_k$ – random aggregate insurance claims; z_k – random claims with distribution function $\overline{F}_{t_k}(\cdot)$ at time moments t_k; N_t – a number of random claims, that happened before time t; c – the aggregate premium per unit of time; $d(\cdot)$ – a function that expresses the intensity of the payment of dividends and of other payments basing on current reserves, $0 \le d(\cdot) \le c$. Function $d(\cdot)$ is interpreted as positional strategy of dividends control. For example

$$d(x) = \begin{cases} a\big(x - b(x)\big), & x \ge b(x), \\ 0, & x < b(x), \end{cases}$$

where $b(\cdot)$ is some monotonically increasing function, that is called dividend barrier, a is a part of the capital, allocated to non-production purposes. In the classical Cramer-Lundberg model [1] (with subtracting constant dividends d, $0 \le d \le c$) the capital evolution equation is as:

$$x_t = u + (c-d)t - \sum_{k=1}^{N_t} z_k,\quad 0 \le t \le T, \qquad (2)$$

where $\{z_k, k = 1,2,...\}$ are independent realizations of random claims with common distribution function $F(\cdot)$ and mean μ, N_t is an integer-valued random variable having Poisson distribution with parameter α (claims arrival time rate in the exponential distribution). In practice, the financial conditions of a company are recorded at discrete moments of time, e.g. quarterly. In such a case, a mathematical model of stochastic evolution of reserves x_t of an insurance company is the following:

$$\begin{aligned} x_{t+1} &= x_t + c_t - x_t - d_t = \\ &= u + \sum_{i=1}^{t} c_i - \sum_{i=1}^{t} x_i - \sum_{i=1}^{t} d_i, \end{aligned} \qquad (3)$$

where $t = 0,1,...,T-1$ – discrete time parameter; $x_0 = u \geq 0$ – a seed capital (reserve); x_t – current insurance reserve at time t; c_t, d_t, x_t – total quarterly premiums, dividends and insurance claims for period t, respectively. The distribution of x_t can be estimated basing on insurance statistical data [9].

Let us consider ruin probability $\psi(\cdot) = \Pr\{\inf_{0\le t\le T} x_t < 0\}$ as a function of the process parameters. This probability can be used as a measure of risk for managing an insurance company. For example, the ruin probability (in the infinite time horizon) in the classical model of an insurance company (2) with exponential claim distribution is as follows [2]:

$$\psi(u,c,d,\alpha,\mu) = \begin{cases} \dfrac{1}{1+\rho}\exp\left(-\dfrac{\rho u}{(1+\rho)\mu}\right), & \rho > 0, \\ 1, & \rho \le 0, \end{cases} \qquad (4)$$

where $\rho = (c-d)/(\alpha\mu) - 1$. In this case the dependence $\psi(u,c,d,\alpha,\mu)$ is known explicitly. In a more general model (1), it can only be obtained by Monte Carlo simulations. Equation (4) in this article is used for testing and tuning the Monte Carlo method. In addition to the ruin probability one may be interested in the dispersion of capital and collected dividends at some moment of time, their average values and variances at that time, and in the dependence of these quantities on a variety of parameters.

Parallel statistical simulation method (Monte-Carlo)

The idea of the method is to simulate in parallel a large number N of trajectories of the evolution of a company reserve x_t in the time interval $[0,T]$ for given parameters (u,c,d,α,μ) and to calculate a share of survived trajectories $p_N(t)$ by the time t. We also calculate the average net income (collected dividends):

$$D_N(T) = (1/N)\sum_{n=1}^{N}\int_0^{\tau_n} d(x_s^n)\,ds,$$

where $\{x_s^n, 0 \le s \le \tau_n\}$ is the trajectory of risk process (1), (2) or (3) in n-th simulation; τ_n is the moment of ruin or $\tau_n = T$, if the n-th trajectory survives until time T. During paths simulation computing cores do not communicate, and once the simulation cycle is complete, information is passed to one core and indicators $p_N(u,c,d,\alpha,\mu,T)$ and $D_N(u,c,d,\alpha,\mu,T)$ are built as a function of a given parameter. The simulation results are displayed in "variable parameter - ruin probability" plane and in the "risk - return" space i.e. in "ruin probability - collected dividends" plane. The accuracy of the Monte Carlo method can be estimated by means of exponential Hoeffding inequality:

$$\Pr\left\{\left|p_N(u,c,d,\alpha,\mu,T)-\psi(u,c,d,\alpha,\mu,T)\right|\ge\delta\right\}\le 2e^{-N\delta^2/2},$$

whence (10^{-k})-confidence interval for $\left|p_N(u,t)-\psi(u,t)\right|$ after N simulations is equal to:

$$\delta_k(N)=\sqrt{2\left(k\ln 10+\ln 2\right)}\Big/\sqrt{N}\,.$$

Software implementation of a parallel Monte Carlo simulation method on GPU

To perform simulations, insurance modeling software system "RMS 0.2" was developed. Graphical interface of the system (Fig. 1, 2) allows studying the dependence of a utility function (for example, the value of collected dividends or the residual capital) on different control parameters. One can also visualize the dependence of ruin probability (used as a measure of risk) on parameters. The interface is built on .NET technology and the calculation core was coded in CUDA C.

The following features were implemented in a current version:

model selection (classical or discrete-time model);

standard statistical data analysis;

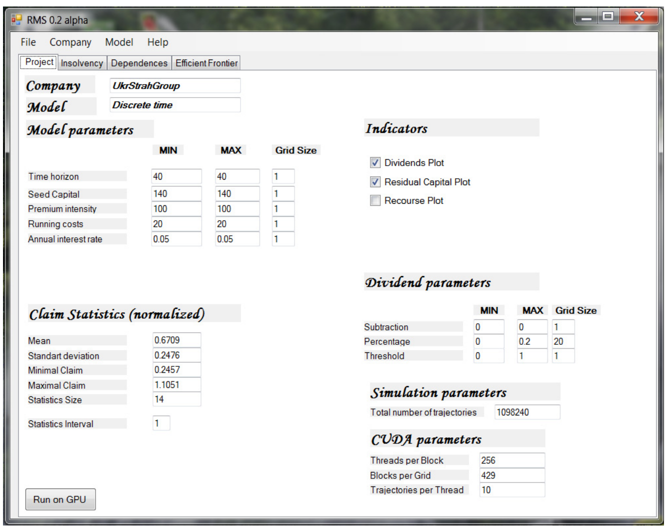

Fig.1. The window of parameters setting in RMS 0.2.

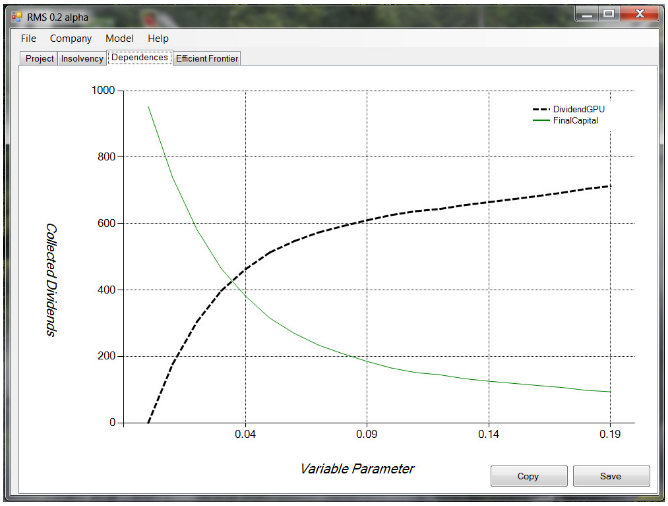

Fig.2. The displaying simulation results.

setting reference intervals and grid sizes for variable parameters;

setting a number of simulations;

setting parameters for parallel computing (load balancing between cores);

project saving and loading (both data and model parameters);

building and visualizing ruin probability, total dividends and the residual capital dependences on any of the model parameter;

displaying simulation results (efficient frontier) in the plane "probability of ruin - total dividends";

printing and saving results.

The system was tested on real world statistical data for specific companies, obtained by processing data from the Ukrainian journals "Strahova sprava" and "Insurance TOP". An example of such data is shown in Fig.3.

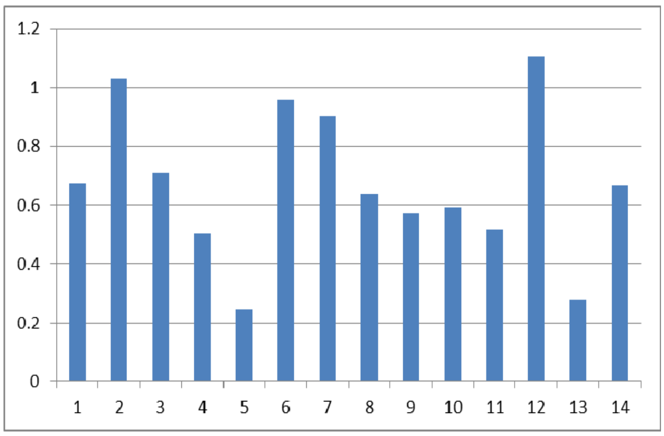

Fig. 3. The empirical distribution of normalized quarterly claims.

For implementation of parallel Monte Carlo method the key problem is generation of a large number of independent long sequences of independent random numbers. In our case, this problem is solved by using a pseudo-random number generation library CUDA CURAND [8]. Experiments have been conducted with single and double precision representations of numeric data.

Numerical results

The problem of finding ruin probability and the net income of the insurance company on a finite time interval $[0,T]$ is considered. For the test purposes the classical Cramer-Lundberg model (2) with exponentially distributed values of claims and its time discretization (3) were used, as the model (2) admits a simple analytical solution (4) for the ruin probability. Numerical experiments were performed on the following equipment - AMD Athlon 64 3200 + 1.5Gb Ram, graphics accelerator - Nvidia GeForce GTX 560 2Gb using NVIDIA CUDA 4.2.

1. GPU acceleration. The comparison of elapsed time on CPU and GPU for estimation of the ruin probability for process (2) with parameters $T=100$ $c=1$, $d=0$, $\alpha=\mu=1$, $u=1$, was made. The observed acceleration when performing calculations on GPU is about one or two orders of magnitude (Fig 4.).

2. The accuracy and the evaluation of small ruin probability on the GPU. The comparison of estimate for ruin probability of process (2) on time interval $T=100$ with its exact value on infinite time interval $T=\infty$ that was obtained by

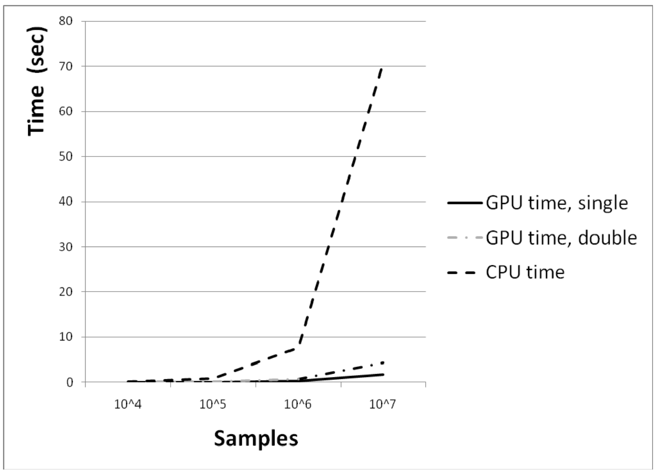

Fig. 4. Comparison of computing speed.

the formula (4), parameters $c=2$, $\alpha=\mu=1$, $u=10$ was conducted, see Tab. 1. This experiment shows that by means of GPU one can accurately evaluate small ruin probabilities in a real time.

Tab. 1. The accuracy and computation time for ruin probability.

Number of trajectories	10^6	10^7
Time (in milliseconds) single precision.	199	1959
Time (in milliseconds) double precision.	581	5019
Relative error, single precision%	3.35	0.654401
Relative error, double precision%	1.03	0.322772
Theoretical ruin probability on infinite time interval	0.003369	

3. Studying ruin probability and net income dependence on parameters by using GPU. Using insurance modeling system RMS 0.1, one can calculate ruin probability and net income (total collected dividends) as functions of any parameter of the model. To do this, one need to specify the minimum and maximum values of varied paramter and the number of intermediate values of the parameter. Reduction of computational time is achieved through massive parallelism of statistical simulation method by GPU. Fig. 5 shows (dotted line) the ruin probability $\psi(u,c,d,\alpha,\mu,T)$ for model (2) as a function of claim arrival intensity $\mu\in[0.5,2.2]$. The solid line shows the exact value of the ruin probability for $T=\infty$ according to formula (3). Other parameters values are: $u=10$, $c=2$, $d=0$, $\alpha=1$, $T=100$. A discrepancy between ruin probability estimate and the exact value appears due to the finite time interval $T=100$ in case Monte-Carlo method.

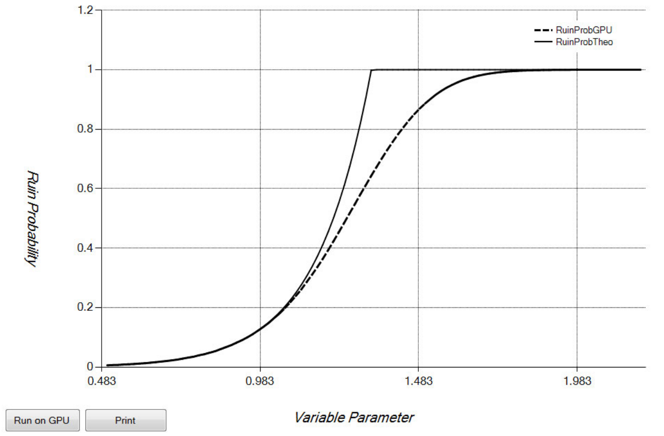

Fig. 5. Ruin probability as a function of claim arrival intensity.

Figs. 6, 7 show the simulation results for process (1) with dividend strategy

$$d(x)=\begin{cases}0, & x<b,\\ d', & x\ge b,\end{cases}$$

where dividend barrier $b \in [5,50]$ is a variable parameter. The rest of the model parameters have the following values: $c = 2$, $\alpha = 2$, $u = 10$, $\mu = 1.5$, $T = 100$, $d' = 1$.

In Fig. 8 the ruin probability and collected dividends are displayed on the same graph as a implicite functions of common variable parameter b.

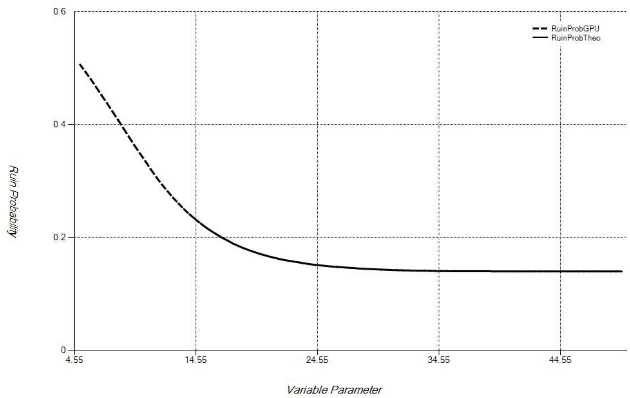

Fig.6. Ruin probability as a function of dividend barrier b.

Conclusions

The usage of graphical accelerators allows performing actuarial calculations by means of Monte Carlo method with an acceptable relative

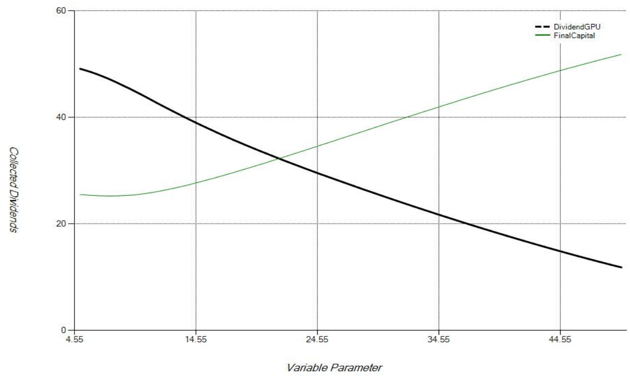

Fig.7. Dependence of the collected dividends and of the residual capital on the threshold b.

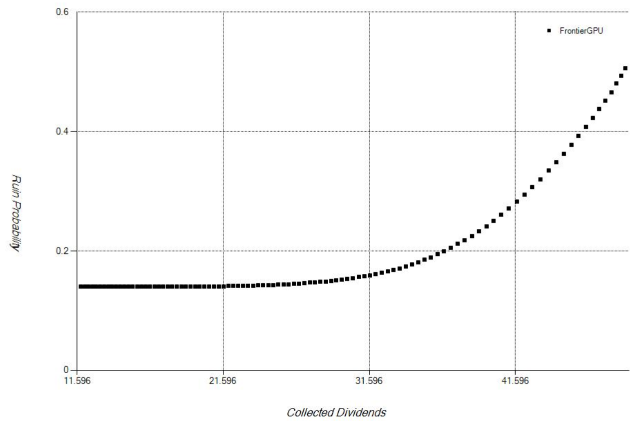

Fig.8. Results of simulation in the plane "profit-risk"

accuracy 1% for ruin probabilities of order 10^{-3} in practically real time. One should take into account that the speed of the Nvidia GeForce 400/ GeForce 500 graphics cards (Fermi architecture) is approximately 8 times faster in performing single-precision calculations compared to double-precision ones. Thus practicality of performing double precision calculations has to be analyzed in every particular case.

The developed system of actuarial modeling RMS 0.2 gives an opportunity for the company risk manager to predict effects of policy changing by building and analyzing ruin probability and net income dependences on any model parameter.

References

1. Norkin B.V. On actuarial calculations by means of graphical accelerators // Proceedings of the International conference «High performance computations» (HPC-UA 2012, Ukraine, Kyiv, 8-10 October, 2012). – Kyiv: National Academy of Sciences of Ukraine, 2012. – P.268-274. (In Russian).
2. Leonenko N.N., Mishura Yu.S., Parhomenko Y.M., Yadrenko N.Y. Teoretiko-ymovirnisni ta statistichni metody v ekonometritzi ta finansoviy matematytsi (Theoretical probabilistic and statistical methods in econometrics and financial mathematics). – Kyiv: Informtechnika, 1995. – 380p. (In Ukrainian).
3. Kaufmann R., Gadmer A., Klett R. Introduction to dynamic financial analysis // ASTIN Bulletin. – 2001. – Vol. 31. – No. 1. – P. 213-249.
4. Norkin B.V. Mathematical models for insurance business optimization // Cybernetics and Systems Analysis. – 2011. – Vol. 47. – № 1. – P. 117-133.

5. Norkin B. Parallel computations in insurance business optimization //Proceedings of the 1-st International Conference on High Performance Computing. October 12-14, 2011, Kyiv, Ukraine. – P. 33-39.
6. Norkin B.V. Rasparalelivanie metodov otzenki riska bancrotstva strahovoi companii (Parallelization of methods for evaluation of ruin probability of an insurance company) // Teoriya optimalnyh rishen. – Kyiv : Institute of Cybernetics, 2010. – P. 33-39. (In Russian).
7. Norkin B.V. O veroyatnosti razoreniya upravlyemogo protzessa autoregressii (On the probability of ruin of a controled auto-regression process) // Kompyutenaya matematika. – Kyiv: Institute of Cybernetics, 2011. – P. 142-150. (In Russian).
8. Haizhen Wu. Parallel Computing Using GPUs. - March 1, 2011. http://ecs.victoria.ac.nz/twiki/pub/EResearch/EcsTeslaResource/Parallel.Computing.Using.Graphics.Cards.pdf
9. Boreskov A.V., Harlamov A.A., Markovskiy N.D. et al. Paralelnye vychisleniya na GPU. Arhitectura i programnaya model CUDA (Parallel computations on GPU. Architecture and program model CUDA). – Moscow: Publication of Moscow University, 2012. – 336 p. (In Russian).
10. Zaletov A.I. Strahovanie v Ukraine (Insurance in Ukraine). – Kyiv: Mezhdunarodnaya agentziya «BeeZone», 2002. – 452 p. (In Russian).
11. NVIDIA, CUDA CURAND Library, 2010.