# Accelerated Variance Reduction Methods on GPU

Chuan-Hsiang Han
Department of Quantitative Finance,
National Tsing-Hua University,
Hsinchu, Taiwan, R.O.C.
chhan@mx.nthu.edu.tw

Yu-Tuan Lin
Institute of Mathematics
Academia Sinica
Taipei, Taiwan, R.O.C.
ytlin@math.sinica.edu.tw

***Abstract*—Monte Carlo simulations have become widely used in computational finance. Standard error is the basic notion to measure the quality of a Monte Carlo estimator, and the square of standard error is defined as the variance divided by the total number of simulations. Variance reduction methods have been developed as efficient algorithms by means of probabilistic analysis. GPU acceleration plays a crucial role of increasing the total number of simulations. We show that the total effect of combining variance reduction methods as efficient software algorithms with GPU acceleration as a parallel-computing hardware device can yield a tremendous speed up for financial applications such as evaluation of option prices and estimation of joint default probabilities.**

***Keywords—variance reduction; GPU acceleration; option pricing; default probability estimation***

## I. INTRODUCTION

Major computational methods [17] in financial applications include tree (or lattice) method, numerical partial differential method, (fast) Fourier transform method, and Monte Carlo method. Unlike others being deterministic based, Monte Carlo method is a probabilistic-based simulation method. By virtue of "free from the curse of dimensionality," Monte Carlo method has been widely employed for solving complex and/or high-dimensional problems.

The quality of a Monte Carlo estimator is often measured by standard error, which is defined by standard deviation of the underlying random variate divided by the square root of the total number of simulations. To reach a high-level accuracy for Monte Carlo method, variance reduction methods have been developed for providing efficient algorithms [15]. Control variate and importance sampling are perhaps mostly used techniques among variance reduction methods. Control variate represents a transformation by addition, while importance sampling represents a transformation by multiplication. Thus standard errors of Monte Carlo estimators can be dramatically reduced by software algorithms; namely, variance reduction.

First manufactured by Nvidia in 1999, a graphics processing unit (GPU) has a highly paralleled hard-ware structure, which is designed for computer graphics rendering. A modern GPU has been designed to accelerate computations for scientific, engineering, and financial applications. GPU appeals to Monte Carlo method because massive parallelism can be exploited. It provides an alternative to largely increase the total number of simulations. Thus standard errors of Monte Carlo estimators can be effectively reduced by such hardware device-GPU.

From the perspective of financial applications, we are particularly interested in solving two kinds of problems. They include derivatives evaluation and risk management. Both are essential to the modern operation of financial institutions [14] because they are associated with two major risks in finance. One is the market risk and the other is credit risk. Option pricing is a core in the problem of derivatives evaluation. One specific European-style option pricing problem under the stochastic volatility model [6] will be explored. Estimating the probability of default is fundamental for credit risk management. One specific probability estimation of joint default under copula models [2] will be studied.

Regarding to those evaluation problems, i.e., the option pricing and the probability of joint default, by Monte Carlo simulations, their corresponding variance reduction methods are often different. The option pricing problem will be solved by (martingale) control variate, while the probability estimation of joint default will be solved by importance sampling. Both algorithms have been proved to be asymptotically optimal [1] in the sense of "asymptotical zero variance." That is, those Monte Carlo estimators associated with both variance reduction algorithms admit zero variance in the limiting situation. Given such efficient algorithms, the additional effect of GPU becomes an interesting object to explore. The goal of this paper is to examine the total effect of combining variance reduction with enlargement of simulation size by GPU for Monte Carlo estimators under these two financial applications.

Our numerical experiments demonstrate tremendous speed up obtained from the combination of variance reduction methods under GPU computing. GPU computing refers to the computer implementation by the programming language CUDA (Compute Unified Device Architecture). Section 3 records variance reduction ratios ranging from 65 to 3844 and GPU speed-up ratios ranging from 18 to 249. If the total effect of variance reduction on GPU acceleration is measured by the square of standard error multiplied by the execution time. We shall see significant figures such as 6057 (option pricing) and 68340 (joint default probability estimation) on the performance of variance reduction methods accelerated by GPU in our numerical experiments.

Efficient algorithms and parallel devices provide a promising framework for solving complex problems.

According to these results, one can further reach out more practical applications in finance. For example, martingale control variate accelerated by GPU, used to solve the option pricing problem, can be applied to solve the model calibration of implied volatility surface [11,13]. Importance sampling accelerated by GPU, used for joint default probability estimation problem, can be extended to the (nonlinear) portfolio default probability estimation problem [8].

The organization of this paper is the following. Section 2 introduces two customized variance reduction methods including martingale control variate and importance sampling. These are efficient algorithms proved to have asymptotical zero variance, and used for solving the option pricing problem and the joint default probability estimation problem, respectively. Section 3 demonstrates numerical results of GPU acceleration on those efficient algorithms. We conclude on Section 4.

## II. Two Variance Reduction Methods: Martingale Control Variate and Importance Sampling

### A. Martingale Control Variate Method: Option Pricing

In computational finacne, perhaps the most successful implementation of control variate is to evaluate continuous-time arithmetic-average Asian options. Kemna and Vorst [16] adopted a discounted counterpart geometric-average Asian option payoff less its price as a control. This method works well for two reasons. First, the correlation between the arithmetic-average and the geometric-average is high. Second, the counterpart geometric-average Asian option price has a closed-form solution. Han and Lai [10] generalized this approach to a dynamic setting termed martingale control variate, which can be used to evaluate a larger class of options such as American option [4], some exotic options [5,9], and a larger class of financial models such as multi-factor stochastic volatility models [7].

We consider a European-style option pricing problem under the one-factor stochastic volatility model. This is a typical class of models that are often used for model calibration to implied volatility surface [11,13]. Under a risk-neutral probability measure $P^*$, the option pricing problem given the payoff function $H(x)$ is defined by

$$P\left(t,S_t,Y_t\right)=E^*\left[e^{-r(T-t)}H\left(S_T\right)\mid S_t,Y_t\right]. \qquad (1)$$

The payoff function is termed a call payoff if $H(x)=\left(x-K\right)^+=\max\left\{x-K,0\right\}$, while it is termed a put payoff if $H(x)=\left(K-x\right)^+=\max\left\{K-x,0\right\}$. The basic Monte Carlo estimator

$$\frac{1}{N}\sum_{i=1}^{N}e^{-rT}H(S_T^{(i)}) \qquad (2)$$

provides an approximation to the option price defined in Equation (1). The super script $(i)$ in Equation (2) indicates the i-th independent sample and $N$ denotes the total number of simulations.

Stochastic volatility models [6] contain a class of continuous-time models in order to capture some stylized effect of volatilities in finance. The asset price dynamics are assumed to satisfy the following stochastic differential equations

$$dS_t=rS_tdt+\sigma_tS_tdW_t^{(0)*}$$

$$\sigma_t=f(Y_t)$$

$$dY_t=\frac{1}{\varepsilon}\left(m-Y_t\right)dt+\frac{\nu\sqrt{2}}{\varepsilon}\left(\rho dW_t^{(0)*}+\sqrt{1-\rho^2}dW_t^{(1)*}\right) \qquad (3)$$

where $S_t$ is the underlying asset price process such as stock price or index price with a constant risk-free interest rate $r$. Its stochastic volatility $\sigma_t$ is driven by a mean-reverting process $Y_t$, which varies on the time scale $\varepsilon$. The vector $\left(W_t^{(0)*},W_t^{(1)*}\right)$ consists of two independent standard Brownian motions. The instantaneous correlation coefficient $\rho$ satisfies $|\rho|<1$. The volatility function $f$ is assumed to be smooth bounded and bounded below away from 0. In addition, $1/\varepsilon$ is the rate of mean reversion, $m$ is the long run mean, and $\nu$ is the long run standard deviation. Its invariant distribution is $N(m,\nu^2)$.

The martingale control variate method proposed by Fouque and Han [3] formulates the following unbiased estimator:

$$\frac{1}{N}\sum_{i=1}^{N}\left[e^{-rT}H(S_T^{(i)})-M_0^{(i)}(P_{BS})\right] \qquad (4)$$

where $P_{BS}\left(s,S_s;\bar{\sigma}\right)$ denotes the Black-Scholes option pricing formula under the constant volatility $\bar{\sigma}$, and the controlled martingale is defined by

$$M_0(P_{BS})=\int_0^T e^{-rs}\frac{\partial P_{BS}\left(s,S_s;\bar{\sigma}\right)}{\partial x}f\left(Y_s\right)S_sdW_s^{(0)*} \qquad (5)$$

The *effective* volatility $\bar{\sigma}$ is defined as the square root of the volatility function $f^2(.)$ averaging with respect to the invariance distribution of the fast varying volatility process $Y_.$ [6].

In this dynamical context, the martingale control $M_0(P_{BS})$ can be understood as the delta hedging portfolio, in which the *delta* $\frac{\partial P_{BS}}{\partial x}$ hedge strategy has been taken to remove the asset price risk. It remains the volatility risk unhedgeable by the delta strategy. The hedging error attributes to the variance of the martingale control variate estimator. The following theorem provides an asymptotic result for variance analysis. A detail account for this type of models can be found on [4,7,9]. The following theorem provides an asymptotic result for variance analysis.

**Theorem 1 [3]** Under the assumptions made above and the payoff function H being continuous piecewise smooth as a call (or a put), for any fixed initial state $(0,x,y)$, there exists a constant $c>0$ such that for $\varepsilon \le 1$

$$Var\left(e^{-rT}H\left(S_T\right)-M_0\left(P_{BS}\right)\right)\le c\varepsilon$$

This theorem guarantees the asymptotical optimality of the martingale control variate. It implies that when the mean-reverting speed $\varepsilon$ goes to zero, this variance reduction algorithm induces a zero variance.

### *B. Importance Sampling: Probability Estimation of Joint Default*

Importance sampling [15] is a crucial technique to estimate probabilities in rare event simulation. Based on the exponential twist for measure change, we study a high-dimensional lower tail probability estimation problem, which is called the joint default probability estimation in finance.

Copula methods are capable of constructing various correlation structures for financial modeling [2]. Archimedian copula and elliptic copula are two major classes of copula. When the problem of joint default probability estimation is concerned, closed-form solutions exist for Archimedian copula, but not elliptic copula. Hence it is an essential task to provide efficient algorithms for estimating probabilities of joint default under elliptic copula including the Gaussian copula and Student's t copula.

Gaussian copula models the joint default by a random vector of multivariate normal. We apply the exponential twisting [15] to derive an importance sampling algorithm. The vector of exponential twisting parameters turns out to satisfy a linear system, which is associated with the covariance matrix defined in the multivariate normal. The variance of this stochastic algorithm is proved to be asymptotically optimal by means of the large deviation theory [1]. In simulation terms, an *efficient* algorithm is defined by its variance being zero asymptotically. See Han and Wu [12] for detailed discussion and its extension to Student's t distribution, which is omitted here to keep the clarity and length constraint of this paper.

A brief review of the importance sampling by exponential twist to estimate the probability $P\left(Z<C\right)=E\left[\mathrm{I}(Z<C)\right]$ is as follows. Suppose that under a probability space $\left(\Omega,F,P\right)$, the multivariate $Z\in R^d$ has a density $f$ with $f(z)>0$ for $z\in R^d$, and its moment generating function denoted by $M_Z(\mu)$ exists, where $\mu=\left(\mu_1,\cdots,\mu_d\right)^T$ denotes a vector of parameters in the generating function. The exponential twist imposes a new density function defined by

$$f_\mu(z)=\frac{\exp(\mu z)f(z)}{M_Z(\mu)} \tag{6}$$

for measure change. Under the new probability measure $P_\mu$ defined from the Rodan-Nykodym derivative

$$dP/dP_\mu=\exp(\mu Z)/M_Z(\mu),$$

the lower tail probability $P(Z<C)$ can be expressed by

$$P_1=E_\mu\left[\mathrm{I}(Z<C)\frac{f(Z)}{f_\mu(Z)}\right] \tag{7}$$

where $E_\mu$ denotes the expectation with respect to the probability measure $P_\mu$.

Let $P_2(\mu)$ denote the second moment of the random variate $\mathrm{I}(Z<C)\dfrac{f(Z)}{f_\mu(Z)}$ under the new measure $P_2(\mu)$ shown in Equation (7). That is,

$$\begin{aligned}P_2(\mu)&=E_\mu\left[\mathrm{I}(Z<C)f^2(Z)/f_\mu^2(Z)\right]\\&=E\left[\mathrm{I}(Z<C)f(Z)/f_\mu(Z)\right]\end{aligned}$$

Substituting the choice of $f_\mu(z)$, defined in Equation (6), into $P_2(\mu)$, we obtain

$$\begin{aligned}P_2(\mu)&=M_Z(\mu)E\left[\mathrm{I}(Z<C)\exp\left(-\mu^TZ\right)\right]\\&\le M_Z(\mu)E\left[\mathrm{I}(Z<C)\exp\left(-\mu^TC\right)\right]\\&\le M_Z(\mu)\exp\left(-\mu^TC\right)\end{aligned} \tag{8}$$

in which we assume all $C$ and $\mu$ are negative numbers for the first inequality to be held. Since the indicator function is bounded above by 1, the second inequality is satisfied.

We intend to minimize the variance of

$$\mathrm{I}(Z<C)f(Z)/f_\mu(Z)$$

over $\mu\in R^d$. This task is reduced to minimize the second moment $P_2(\mu)$ because $P_1$ is $\mu$-independent. It is a challenging problem to solve for the minimizer of $P_2(\mu)$ particularly in high-dimensional cases. When the moment generating function is in exponential form, it may appear that minimizing the logarithm of the upper bound, last inequality in

Equation (8), becomes tractable. Thus, it provides a candidate for importance sampling. According to the first order condition, each partial derivative must be zero to solve for $\mu$ as follows. For $i = 1, \cdots, d$,

$$\frac{1}{M_Z(\mu^*)} \frac{\partial M_Z(\mu^*)}{\partial \mu_i^*} = c_i \ \mu^* = \left(\mu_1^*, \cdots, \mu_d^*\right)^T .$$

In the case of multivariate normal, it turns out these equations can be solved explicitly for $\mu^*$. In fact, the optimal solution $\mu^*$ satisfies this linear equation

$$\Sigma \mu^* = C .$$

This is because the moment generating function of $Z \sim N(0, \Sigma)$ is

$$M_Z(\mu) = \exp\left(\frac{1}{2}\mu^T \Sigma \mu\right).$$

In order to facilitate numerical comparisons, a pseudo algorithm for estimation of a lower tail probability under a centered multivariate normal Z is given below.

**Algorithm 1**: Estimation of lower tail probability $P(Z < C)$ under a centered multivariate Normal $Z$

1. Given the distribution $Z \sim N(0, \Sigma)$ and the lower threshold $C < 0$, compute $\mu^* = \Sigma^{-1} C$
2. For each independent ith replication, $i = 1, \cdots, m$
   (a) Generate $Z^{(i)} = \left(Z_1^{(i)}, \cdots, Z_d^{(i)}\right)^T$ from $N(C, \Sigma)$.
   (b) Evaluate $M_Z(\mu^*) \exp\left(-\mu^{*T} Z^{(i)}\right) \mathrm{I}(Z^{(i)} < C)$.

3. Compute the average of samples generated from (b) in Step 2.

Next theorem analyzes this algorithm rigorously by means of the large deviation theory.

**Theorem 2**.[12] Assume that the scale $\alpha$ is a positive number, each element in the vector $C \in R^d$ is negative, $W \sim N(0, I_d)$, and the lower triangular matrix A satisfies the Cholesky decomposition of a given covariance matrix $AA^T = \Sigma$. We obtain the following asymptotic approximation:

$$\lim_{\alpha\to\infty} \frac{1}{\alpha} \ln P_2\left(\mu^*; \sqrt{\alpha} C\right) = 2 \lim_{\alpha\to\infty} \frac{1}{\alpha} \ln P_1\left(\sqrt{\alpha} C\right) = -C^T \Sigma^{-1} C$$

Note that we have scaled the default threshold from $C$ to $\sqrt{\alpha} C$ so that the asymptotic analysis can be conducted. Moreover, from the first moment approximation $P_1\left(\sqrt{\alpha} C\right) \approx e^{-\frac{\alpha}{2} C^T \Sigma^{-1} C}$ and the second moment approximation $P_2\left(\mu^*; \sqrt{\alpha} C\right) \approx e^{-\alpha C^T \Sigma^{-1} C}$, it is easy to see that variance of the importance sampling estimator is approximated zero for large scale $\alpha$.

That implies that the importance sampling scheme

$$P_1\left(\sqrt{\alpha} C\right) =$$
$$E_{\mu^*}\left[\mathrm{I}(X < \sqrt{\alpha} C) \exp\left(2\sqrt{\alpha} C^T \Sigma^{-1} X + \alpha C^T \Sigma^{-1} C\right)\right]$$

where $X := AW \sim N(\sqrt{\alpha} C, \Sigma)$ under the probability measure $P_{\mu^*}$ is asymptotically optimal to estimate the lower tail probability, i.e. the joint default probability.

## III. GPU ACCELERATION ON VARIANCE REDUCTION METHODS

According to the aforementioned efficient variance reduction algorithms of martingale control and importance sampling, we further investigate their performance under the computing environment of CPU or GPU. Configuration of executing computers include CPU (Core i7 950, 4-core 3.06 GHz) and GPU (NVIDIA GeForce GTX 690, 3072 CUDA core, 915 MHz).

### *A. Martingale Control Variate on GPU*

To facilitate the computation of the option price defined in Equation (1) as a conditional expectation, two Monte Carlo estimators are used for approximations. They include the basic Monte Carlo (BMC in short) estimator, shown in Equation (2), and the martingale control variate (MCV in short) estimator, shown in Equation (4). We adopt the Euler scheme [18] for discretizing the pricing dynamics defined in Equation (3) as well we the martingale control $M_0(P_{BS})$ in Equation (5).

The payoff function of a European option is $H(x) = (x - K)^+$. The chosen model parameters include the risk-free interest rate $r = 0.05$, the volatility time scale $\varepsilon = 5$, the long-run mean $m = 2$, the long-run standard deviation $\nu = 3$, and the correlation coefficient $\rho = 0.7$. Variables of the option contract include the strike price is $K = 90$ and the time to maturity is $T = 1$. Given the current time being $t = 0$, all initial conditions of those stochastic differential equations include $S_0 = 100$ and $Y_0 = 2$.

For executing Monte Carlo simulations, the step size of time discretization is set $\Delta t = 0.01$ and the total number of simulations is set $N = 10{,}000$ in this numerical experiment.

Table 3.1 Results of option pricing by Monte Carlo methods under a single-factor stochastic volatility model

| | | GPU | CPU | Speed up |
|---|---|---|---|---|
| **BMC** | Mean | 17.1324 | 16.8473 | |
| | SE | 0.0328 | 0.0305 | |
| | Time | 0.162 (s) | 40.47 (s) | **X249** |
| **MCV** | Mean | 16.8030 | 16.8072 | |
| | SE | 0.0037 | 0.0037 | |
| | Time | 0.454 (s) | 71.54 (s) | **X157** |
| **Accuracy** | Variance Reduction Ratio | **X76** | **X65** | |

BMC stands for the basic Monte Carlo method defined in Equation (2). MCV stands for martingale control variate method defined in Equation (4).

When the total effect of variance reduction on GPU acceleration is defined as the execution time multiplied by the square of standard error, we shall see that the combination of GPU and MCV versus the combination of CPU and BMC results in a total reduction of 6057 times.

Randomized quasi Monte Carlo methods are shown to have the best convergence rate among Monte Carlo methods. Its implementation on MCV has been conducted by Han and Lai [9]. Due to the current limitation of generating quasi random sequences such as Sobol's sequence on GPU, we comment that there is a potential to overwrite the total effect mentioned above in the future when a large scale of Sobol's sequence in high dimension can be generated from CUDA.

### *B. Importance Samping on GPU*

The basic Monte Carlo estimator for the joint default probability is

$$P\left(Z < C\right) \approx \frac{1}{N}\sum_{i=1}^{N} \mathrm{I}\left(Z^{(i)} < C\right), \qquad (9)$$

where the super script $(i)$ denotes the i-th independent sample. The setup of our numerical experiment is the following: the total number of simulations $N = 2{,}000{,}000$ , $Z$ is a 40 dimensional multivariate normal with mean zero and the variance-covariance matrix being one's in the diagonal and 0.5's off-diagonal, and each entry of the default threshold vector C is -2 homogeneously. The importance sampling estimator for the joint default probability can be implemented by Algorithm 1. Numerical results are illustrated in Table 3.2.

When the total effect of variance reduction on GPU acceleration is defined as the execution time multiplied by the square of standard error multiplied, we shall see that the combination of GPU and IS versus the combination of CPU and BMC results in a total reduction of 68,340 times. As for Student's t distribution being a fat-tail distribution among the elliptic copula, similar numerical results can be obtained but they are skipped here. See Han and Wu [12] for more discussions and their comparisons with Matlab codes *mvncdf.m* and *mvtcdf.m*.

Table 3.2 Result of Monte Carlo Simulation under multivariate normal distribution (sample size = 2M, d=40.)

| | | GPU | CPU | Speed up |
|---|---|---|---|---|
| **BMC** | Mean | 1.50e-6 | 1.50e-6 | |
| | SE | 8.7e-7 | 8.7e-7 | |
| | Time | 0.05 (s) | 1.77 (s) | **X36** |
| **IS** | Mean | 2.00e-6 | 2.01e-6 | |
| | SE | 1.40e-8 | 1.39e-8 | |
| | Time | 0.10 (s) | 1.77 (s) | **X18** |
| **Accuracy** | Variance Reduction Ratio | **X3861** | **X3918** | |

BMC stands for the basic Monte Carlo method defined in Equation (9). IS stands for importance sampling defined in Algorithm 1.

## IV. Conclusion

The total effect of variance reduction methods and GPU computing is significant. Speed up rates of 6057 and 68,340 are documented for solving problems of option pricing by means of martingale control variate on GPU and joint default probability estimation by importance sampling on GPU, respectively. These efficient algorithms show a great potential to further investigate more practical applications such as Monte Carlo calibration to implied volatility surface, the portfolio optimization under a Value-at-Risk constraint, etc. We leave these as future research.

a.


## Acknowledgment

C.-H. Han thanks a student team from the SCOPE lab from Department of Computer Science, National Tsing-Hua University, for generating numerical results of importance sampling on GPU, and Nvidia-NTHU joint lab on computational finance.


## References


[1] J. A. Bucklew, Introduction to rare event simulation, Springer, 2003.

[2] U. Cherubini, E. Luciano, and W. Vecchiato, Copula Methods in Finance, Wiley, 2004.

[3] J.-P. Fouque and C.-H. Han, "A martingale control variate method for option pricing with stochastic volatility," ESAIM: Probability and Statistics, vol. 11, pp. 40-54, 2007.

[4] J.-P. Fouque and C.-H. Han, "Asymmetric variance reduction for pricing American options," Modelling and Numerical Methods in Finance,



This work is supported by NSC -102-2115-M-007-002 from the first author.

Volume 15: Special Volume (Handbook of Numerical Analysis), Editors A. Bensoussan, Q. Zhang, and Ph. Ciarlet, Elsevier 2009.

[5] J.-P. Fouque, C.-H. Han, and Y. Lai, "Variance reduction for MC/QMC methods to evaluate option prices," Recent Advances in Financial Engineering (Proceedings of the 2008 Daiwa International Workshop on Financial Engineering), World Scientific Co, 2009.

[6] J.-P. Fouque, G. Papanicolaou, R. Sircar, Derivatives in Financial Markets with Stochastic Volatility, Cambridge University Press, 2000.

[7] J.-P. Fouque, G. Papanicolaou, R. Sircar, and K. Solna, Multiscale Stochastic Volatility for Equity, Interest Rate, and Credit Derivatives: Cambridge University Press. 2011.

[8] C. H. Han, "A new importance sampling method for portfolio default probability estimation by high-dimensional embedding," unpublished.

[9] C.H. Han and Y. Lai, "A smooth estimator for MC/QMC methods in finance," Mathematics and Computers in Simulation, vol. 81, pp. 536-550, 2010.

[10] C.H. Han and Y. Lai, "Generalized control variate methods for pricing Asian options," Journal of Computational Finance, Volume 14/Number 2, Winter 2010.

[11] C.H. Han, G. Molina, and J.P. Fouque, "MCMC estimation of multiscale stochastic volatility models with applications," Mathematics and Computers in Simulation, vol. 103, pp. 1–11, September 2014.

[12] C.H.Han and C.-T.Wu, "Efficient importance sampling for estimating lower tail probabilities under Gaussian and Student's *t* distributions," unpublished.

[13] S. Heston, "A closed-form solution for options with stochastic volatility with applications to bond and currency options," Review of Financial Studies, 6(2), pp. 327-343, 1993.

[14] J. Hull, Options, futures, and other derivatives, Eighth Edition, Prentice Hall, 2012.

[15] P. Glasserman, Monte Carlo methods for financial engineering, Springer-Verlag, New York, 2003.

[16] A.G.Z. Kemna and A.C.F. Vorst, "A pricing method for options based on average asset values," Journal of Banking and Finance, 14:113-129, 1990.

[17] J. Kienitz and D. Wetterau, Financial Modeling: Theory, Implementation and Practice with Matlab source, Wiley Finance. 2012.

[18] P. E. Kloeden and E. Platen, Numerical solution of stochastic differential equations, Corrected edition. Springer, 1992.