University of
Strathclyde
Glasgow

Strathprints Institutional Repository

Higham, Desmond and Roj, Mikolaj (2012) *Computing mean first exit times for stochastic processes using multi-level Monte Carlo.* In: Proceedings of the 2012 Winter Simulation Conference. IEEE.

http://strathprints.strath.ac.uk/

COMPUTING MEAN FIRST EXIT TIMES FOR STOCHASTIC PROCESSES USING MULTI-LEVEL MONTE CARLO

Desmond J. Higham
Mikolaj Roj

Department of Mathematics and Statistics
University of Strathclyde
Glasgow, G1 1XH, UK

ABSTRACT

The multi-level approach developed by Giles (2008) can be used to estimate mean first exit times for stochastic differential equations, which are of interest in finance, physics and chemical kinetics. Multi-level improves the computational expense of standard Monte Carlo in this setting by an order of magnitude. More precisely, for a target accuracy of TOL, so that the root mean square error of the estimator is $O(\text{TOL})$, the $O(\text{TOL}^{-4})$ cost of standard Monte Carlo can be reduced to $O(\text{TOL}^{-3}|\log(\text{TOL})|^{1/2})$ with a multi-level scheme. This result was established in Higham, Mao, Roj, Song, and Yin (2013), and illustrated on some scalar examples. Here, we briefly overview the algorithm and present some new computational results in higher dimensions.

1   INTRODUCTION

We consider a system of stochastic differential equations (SDEs)

$$dX(s) = b(X(s))ds + \sigma(X(s))dW(s), \tag{1}$$

with deterministic initial condition $X(0) = x$. Here,

- X takes values in $\mathbb{R}^d$, $b : \mathbb{R}^d \to \mathbb{R}^d$, and $\sigma : \mathbb{R}^d \to \mathbb{R}^{d \times d_1}$,
- $W = \{W(t) : t \geq 0\}$ is a standard d_1-dimensional Brownian motion, and
- $(\Omega, \mathscr{F}, \mathbb{P}, \mathscr{F}_t)$ is a complete, filtered probability space satisfying the usual conditions.

Given a finite time T and an open set $O \subset \mathbb{R}^d$, the *stopped exit time*, τ, is the first time at which $X(s)$ leaves the open set O, or T if this is smaller. We wish to approximate the expected value of this random variable.

We focus here on algorithms based on stochastic simulation—that is, we approximate the mean stopped exit time by computing pathwise samples and combining them into a Monte Carlo sample average. This approach has proved popular (Andersen 2000; Boyle, Broadie, and Glasserman 1997; Longstaff and Schwartz 2001; Mannella 1999) due to its simplicity and its ability to deal with high dimensions and complicated boundaries. In Higham, Mao, Roj, Song, and Yin (2013) a multi-level version of the Monte Carlo approach was proposed and analyzed, based on the ideas of Giles (2008) and Heinrich (2001). Our aim here is to give an overview of the approach and present further computational tests that back up the analysis.

Section 2 briefly describes the method, in relation to standard Monte Carlo, and its computational complexity, and section 3 presents the new computational results.

2 THE MULTI-LEVEL ALGORITHM

We use the basic, fixed stepsize, Euler–Maruyama method to simulate paths of the SDE (1). In the setting of standard Monte Carlo, we let $v^{[i]}$ denote the exit time sample arising from the ith path; that is, the minimum of (a) first time point at which the numerical solution leaves the region, and (b) the finite time limit T. The sample average over N independent paths then has the form

$$\mu = \frac{1}{N}\sum_{i=1}^{N} v^{[i]}.$$

The overall error divides naturally into two parts,

$$\mathbb{E}[\tau] - \mu = \mathbb{E}[\tau - v + v] - \mu = (\mathbb{E}[\tau - v]) + (\mathbb{E}[v] - \mu).$$

The bias: that is, the weak error of the numerical method in terms of its ability to approximate the mean stopped exit time of the SDE. Results in Gobet and Menozzi (2004), Gobet and Menozzi (2007), Gobet and Menozzi (2010) show that this term is $O(h^{\frac{1}{2}})$ for a stepsize h.

The statistical error: which scales like $O(1/\sqrt{N})$ from the perspective of confidence interval width; see, for example, (Glasserman 2004).

Whether we measure computational cost in terms of

- the total number of evaluations of the coefficients $b(\cdot)$ or $\sigma(\cdot)$, or
- the number of pseudo-random number calls to obtain the Brownian increments for Euler–Maruyama,

we arrive at the conclusion that the expected computational cost of each path is proportional to h^{-1}. Letting TOL denote the target level of accuracy, in terms of confidence interval width, so that the root mean square error is $O(\text{TOL})$, we require $O(h^{\frac{1}{2}}) \sim \text{TOL}$ in order to control the weak error and $1/\sqrt{N} \sim \text{TOL}$ to control the statistical error. The overall expected computational cost of the standard Monte Carlo method therefore scales like $Nh^{-1} = O(\text{TOL}^{-4})$.

The multi-level version uses a range of different stepsizes of the form $h_l = M^{-l}T$, for $l = 0,1,2,...,L$, where M is a fixed integer. (For all examples here we use $M = 4$.) At the most refined level, where $h_L = M^{-L}T$, we make the bias in the discretization method match the target accuracy of $O(\text{TOL})$ by forcing $L = \frac{\log \text{TOL}^{-2}}{\log M}$. Letting $v_l^{[i]}$ denote the Euler-Maruyama stopped exit time for each independent path i using stepsize h_l, our overall approximation then takes the form

$$\mu = Z_0 + \sum_{l=1}^{L} Z_l = \frac{1}{N_0}\sum_{i=1}^{N_0} v_0^{[i]} + \sum_{l=1}^{L}\frac{1}{N_l}\sum_{i=1}^{N_l}\left(v_l^{[i]} - v_{l-1}^{[i]}\right).$$

A key additional fact is that the two samples $v_l^{[i]}$ and $v_{l-1}^{[i]}$ come from the same Brownian path at two different levels of resolution. So generally, while independent Brownian paths are used at different levels l, within any particular level l, for $l > 0$, the pair $v_l^{[i]}$ and $v_{l-1}^{[i]}$ comes from the same path.

It is shown in Higham, Mao, Roj, Song, and Yin (2013) that a particular choice of samples-per-level, N_l, given by (2) below, leads to a method with the required user-specified accuracy of TOL having complexity $O(\text{TOL}^{-3}|\log(\text{TOL})|^{1/2})$. This gives an almost $O(\text{TOL}^{-1})$ improvement over standard Monte Carlo.

We refer to Giles (2008) and the growing amount of follow-up literature for intuition about how the multi-level approach adds value, and to Higham, Mao, Roj, Song, and Yin (2013) for more details about the specific context of mean exit time computation. However, we make the important point that the success of the multi-level method relies on **both the strong and weak convergence properties** of the Euler–Maruyama method, despite the fact that a weak approximation is under investigation. Hence, the crucial step in the analysis was to establish the strong convergence rate of the Euler–Maruyama method in terms of its ability to approximate a hitting time. We recall this result from Higham, Mao, Roj, Song, and Yin (2013).

Theorem 1 Under the following assumptions:

1. **C^2 continuity:** b and σ have two continuous bounded derivatives on O,
2. **Strong ellipticity:** for some $c > 0$, $\sum_{ij}(\sigma(x)\sigma^*(x))_{ij}\xi_i\xi_j > c|\xi|^2$, for all $x \in O$, $\xi \in \mathbb{R}^d$,
3. **Regularity of the boundary:** for $d > 1$, $O \subset \mathbb{R}^d$ is a bounded open set with its boundary ∂O being C^2 smooth,

we have
$$\mathbb{E}\left[|\tau - \nu|^p\right] = O(|h\log(h)|^{\frac{1}{2}}), \ \forall p \geq 1.$$

Here, smoothness of the coefficients is required only on the compact domain O, so that many nonlinear SDEs (which, for example, may fail to have bounded derivatives on the whole of $\mathbb{R}^d$) are covered by the analysis.

3 COMPUTATIONAL RESULTS

In the following subsections we present computational results for two-dimensional Brownian motion, a simple neural network (two-dimensional correlated Brownian motion with drift hitting lines) and a first-to-default basket of corporate bonds (5-dimensional geometric Brownian motion.) For each example we confirm that the multi-level approximation converges to the correct value and estimate the order of the variance between fine and coarse approximations of stopped exit times. We also record the computational complexity, in comparison with the standard Monte Carlo method. Our aim is to add to the numerical tests in Higham, Mao, Roj, Song, and Yin (2013) by considering SDEs in more than one dimension and problems on a non-compact domain. Example 3.1 fulfills all the assumptions of Theorem 1; examples 3.2 and 3.3 violate the assumption of a non-compact domain. However, we show numerically that the multi-level algorithm performs well in this setting, too.

3.1 Two-dimensional Brownian Motion

We first simulate two-dimensional Brownian motion starting at the origin, with boundary given by a ball of radius R. In the notation of (1) we have $d = 2$, $b \equiv 0$ and $\sigma = I$, where I denotes the identity matrix. The finite cutoff time is set to $T = 1$ and simulations are performed for three different radius sizes. By varying the radius we check whether the multi-level algorithm performs well when the majority of sample paths leave the domain in the fixed time T and also when they stay in the domain within this time. The estimated stopped exit time value is compared against a reference value, which in this case was obtained using Monte Carlo simulations with $N = 5 \times 10^7$ samples and a stepsize $h = 10^{-4}$. The results are presented in Table 1. We observe that a more stringent accuracy requirement decreases the error in the multi-level estimate, as expected.

In a separate test, we then fix $R = 1$ and check the convergence behavior. In the left picture of Figure 1 we plot on a log-log scale the accuracy obtained by the multi-level algorithm as a function of TOL for TOL $= 10^{-3}$, 5×10^{-3}, 10^{-2} and 5×10^{-2}. We see that the algorithm produces an error that scales like TOL. A line with a slope 1 is included for reference.

In the right picture of Figure 1 we plot the quantity $\log(\text{Var}[\nu_l - \nu_{l-1}])/\log(M)$ over different levels, for a user-specified accuracy of TOL $= 0.001$. A least squares fit for the slope produces $q = -0.5525$ with a residual of 0.1884. A line with a slope $-\frac{1}{2}$ is also included for reference. This agrees with the estimate in Higham, Mao, Roj, Song, and Yin (2013), which forms a key part of the complexity analysis.

Finally, we compare the computational complexity of standard Monte Carlo with the multi-level version in Figure 2. The computational cost of the multi-level version is measured as

$$\text{Cost}_{\text{MLMC}} := \left(N_0 + \sum_{l=1}^{L} N_l M^l\right)\frac{d}{T}\mathbb{E}[\tau],$$

Table 1: Reference solution was obtained using Monte Carlo simulations, $N = 5 \times 10^7, h = 10^{-4}$. Values in parentheses indicate the half-width of the 95% confidence interval.

Radius	Reference value	TOL	MLMC	Absolute error %
$R = 0.5$	0.1279 (0.00003)	0.01	0.1317	2.97
		0.005	0.1354	5.86
		0.001	0.1294	1.17
$R = 1.0$	0.4737 (0.00008)	0.01	0.5037	6.33
		0.005	0.4807	1.48
		0.001	0.4758	0.44
$R = 1.5$	0.7830 (0.00008)	0.01	0.8454	7.97
		0.005	0.7853	0.29
		0.001	0.7847	0.22

where d is the dimension of the approximated process and N_l is the number of paths used on each level, calculated as

$$N_l = \left\lceil 2\text{TOL}^{-2} \sqrt{\text{Var}[v_l - v_{l-1}]M^{-l}} \left(\sum_{l=0}^{L} \sqrt{\text{Var}[v_l - v_{l-1}]M^l} \right) \right\rceil, \quad 0 \le l \le L. \tag{2}$$

We measure the computational cost of the standard Monte Carlo method as

$$\text{Cost}_{\text{stdMC}} := \frac{Nd}{h}\mathbb{E}[\tau],$$

where h is the fixed stepsize such that $h = \text{TOL}^2$ and N is the total number of sample paths, chosen adaptively to obtain the required variance. In Figure 2 we plot on a log-log scale the computational cost as a function of the accuracy TOL. For $\text{TOL} = 2^{-4}$, 2^{-5}, 2^{-6}, 2^{-7} and 2^{-8} we perform 50 multi-level Monte Carlo computations using different initial states of the pseudo-random number generator, and plot the complexity results with green circles. Then we take averages for each accuracy, indicated as black crosses, and compare them with the complexity of standard Monte Carlo, indicated as red squares. A least squares fit performed on the Monte Carlo slope produces $q = -4.2156$ with a residual 1.0966, and on the multi-level Monte Carlo slope gives $q = -3.1688$ with a residual of 1.2891. This is in agreement with the analytical results quoted in Section 1: the standard Monte Carlo complexity of $O(\text{TOL}^{-4})$ and the multi-level Monte Carlo complexity of $O(\text{TOL}^{-3}|\log(\text{TOL})|^{1/2})$.

3.2 Simple Neural Network

We now apply the multi-level Monte Carlo algorithm in a neural network setting. Various models have been considered for the firing of single neurons, but there is an agreement among researchers that if the electrical state of the neural membrane is stated as a single number, which moves towards or away from the firing potential depending on whether the neuron receives excitatory or inhibitory input, respectively, then the time to firing can be estimated by the first hitting time of a certain level for a Brownian motion with drift (Fienberg 1974; Gerstein and Mandelbrot 1964; Iyengar 1985).

Here we are interested in the mean first hitting time of 2-dimensional correlated Brownian motion with drift b,

$$dX_i(s) = b_i ds + \sigma_i dW_i(s),$$

with constant drift $b_1 = 0.1$, $b_2 = 0.2$, constant diffusion coefficient $\sigma_i = 1$, $i = 1, 2$, initial value $X_i(0) = 1$, $i = 1, 2$ and correlation coefficient $\rho = -0.5$, producing the variance-covariance matrix $\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$.
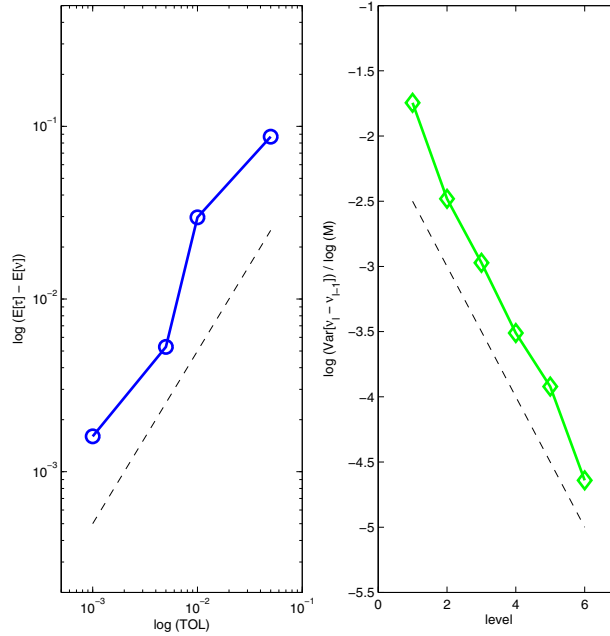
Figure 1: Two-dimensional Brownian motion. Left: weak error of the multi-level algorithm. Right: variance of $v_l - v_{l-1}$ over a sequence of levels.

The Cholesky decomposition of Σ gives $C = \begin{bmatrix} 1 & -0.5 \\ 0 & 0.8660 \end{bmatrix}$, which is then used to simulate correlated Brownian motions. For $i = 1, 2$ we define the stopping times

$$\tau_i = \inf\{t > 0 : X_i(t) \leq B_i\},$$

where we set $B_1 = 0.5$ and $B_2 = 0.25$. The quantity of interest is then $\mathbb{E}\left[(\tau_1 \wedge \tau_2) \wedge T\right]$. In Figure 3 we check the convergence rates of the algorithm and variance. The results are consistent with those in subsection 3.1. A least squares fit for the variance slope gives $q = -0.5025$ with a residual 0.0559.

We also check the computational cost of the algorithm in Figure 4. A least squares fit for the standard Monte Carlo slope produces $q = -3.9492$ with a residual 1.1694, whereas the multi-level method gives $q = -3.2004$ with a residual 0.5913, confirming the theoretical complexity outlined in Section 1.

3.3 First-to-Default Swaps

Finally, we apply the multi-level Monte Carlo algorithm in a financial setting to basket default swaps used in risk management. These financial instruments are derivative securities tied to an underlying basket of corporate bonds or other assets subject to credit risk. A basket default swap gives the protection buyer a type of insurance against the possibility of default in exchange for regular payments made to the protection seller (Chen and Glasserman 2008). These instruments are popular mainly because insuring a basket of assets is usually cheaper than insuring each asset separately. We focus on the example of a first-to-default swap in which the protection buyer is compensated if any asset in the basket defaults, at which time the contract expires (Joro, Niu, and Na 2004).

Without trying to price the protection and value legs at several dates, here we are interested in mean first default time of the first-to-default basket of 5 corporate bonds with maturity $T = 2$ years. We note that it is common to use structural models (see, for example, (Black and Cox 1976)) with a geometric Brownian motion to describe dynamics for the value of the underlying assets (Crouhy, Galai, and Mark
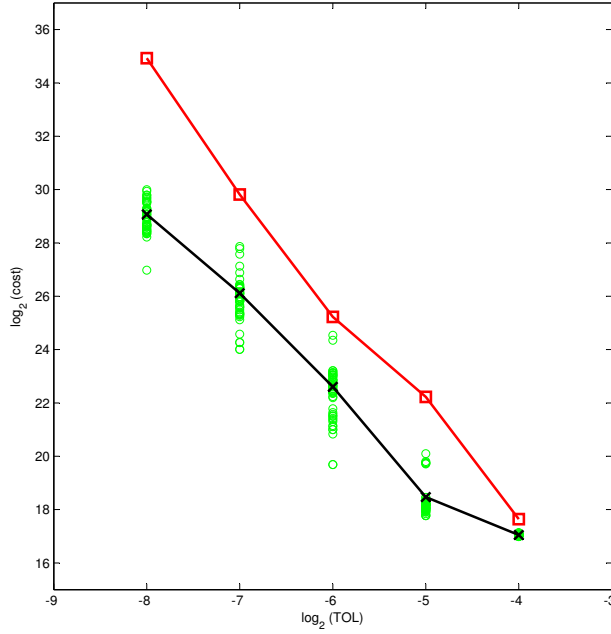
Figure 2: Two-dimensional Brownian motion. Computational cost of standard Monte Carlo (red squares) and multi-level Monte Carlo (green circles). Black crosses indicate averages of multi-level Monte Carlo cost for each accuracy.

2000). Hence, we simulate the underlying processes using a geometric Brownian motion model

$$dX_i(s) = b_i X_i(s)ds + \sigma_i X_i(s)dW_i(s),$$

with constant drift coefficient $b_1 = 0.11$, $b_2 = 0.09$, $b_3 = 0.07$, $b_4 = 0.04$, $b_5 = 0.02$, and constant volatility $\sigma_1 = 0.4$, $\sigma_2 = 0.6$, $\sigma_3 = 0.8$, $\sigma_4 = 0.8$, $\sigma_5 = 1$. The initial value of each company is set to 1, i.e., $X_i(0) = 1$, $i = 1, \ldots, 5$. The lower boundary, which is a default level, is set to $B = 0.1$. We are interested in estimating the quantity $\mathbb{E}\left[(\tau_1 \wedge \tau_2 \wedge \tau_3 \wedge \tau_4 \wedge \tau_5) \wedge T\right]$. We used Monte Carlo simulations with a fixed timestep $h = 10^{-3}$ and $N = 5 \times 10^6$ samples to obtain a reference value of $\mathbb{E}[\tau \wedge T] = 1.2359$ for the mean first default time of the basket. The half-interval width of the 95% confidence interval is 0.0005, making the statistical error negligible.

We use the same format as in the previous two examples. In the right hand picture of Figure 5 a least squares fit for the slope produces $q = -0.6049$ with a residual of 0.5252.

In Figure 6 a least squares fit performed on the Monte Carlo slope produces $q = -4.1018$ with a residual 2.9224, and on the multi-level Monte Carlo slope gives $q = -2.7704$ with a residual of 1.4031. The results are also consistent with the theoretical predictions.
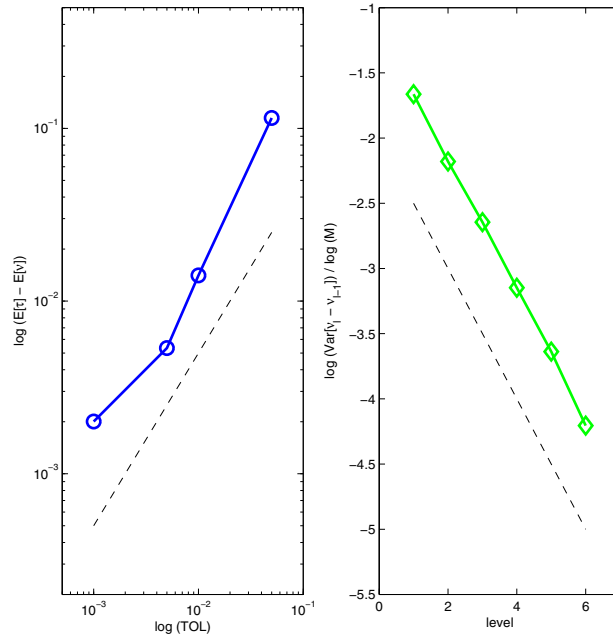
Figure 3: Neural network. Left: weak error of the multi-level algorithm. Right: variance of $v_l - v_{l-1}$ over a sequence of levels.
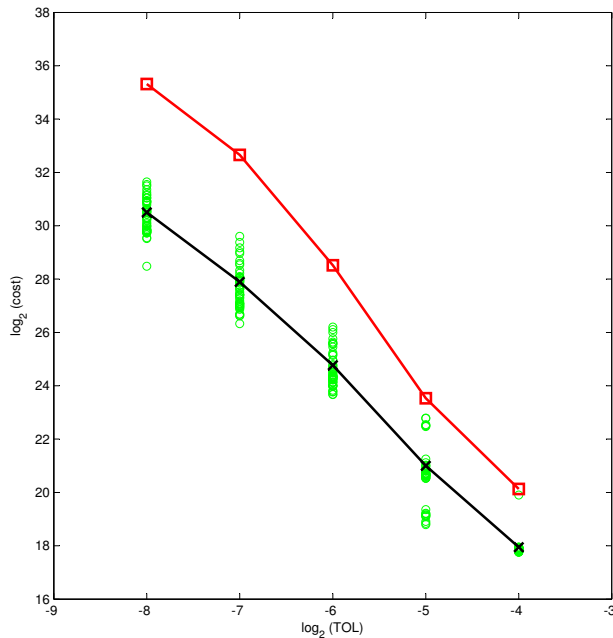

Figure 4: Neural network. Computational cost of standard Monte Carlo (red squares) and multi-level Monte Carlo (green circles). Black crosses indicate averages over 50 simulations of multi-level Monte Carlo cost for each accuracy.
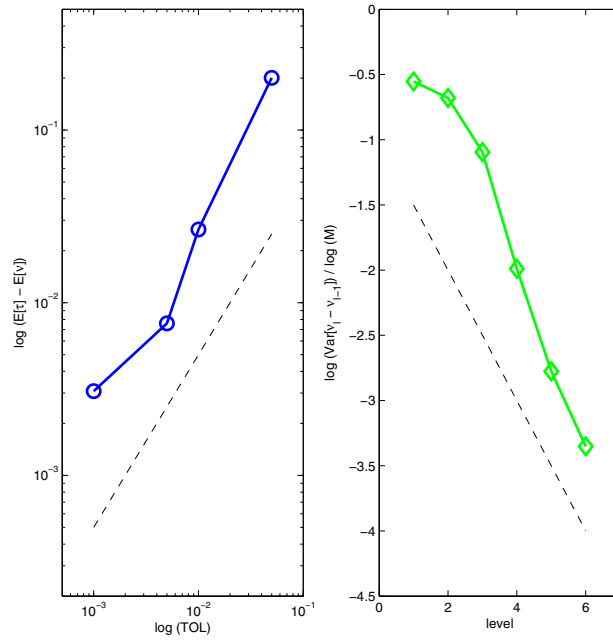
Figure 5: First-to-default. Left: weak error of the multi-level algorithm. Right: variance of $v_l - v_{l-1}$ over a sequence of levels.
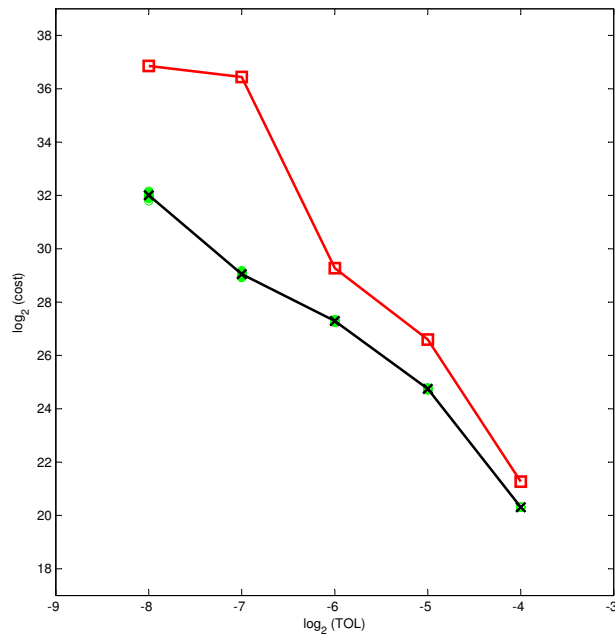

Figure 6: First-to-default. Computational cost of standard Monte Carlo (red squares) and multi-level Monte Carlo (green circles). Black crosses indicate averages over 30 simulations of multi-level Monte Carlo cost for each accuracy.

ACKNOWLEDGMENTS

REFERENCES

Andersen, L. 2000. "A Simple Approach to the Pricing of Bermudan Swaptions in the Multi-Factor Libor Market Model". *Journal of Computational Finance* 3:5–32.

Black, F., and J. Cox. 1976. "Valuing Corporate Securities: Some Effects of Bond Indenture Provisions". *The Journal of Finance* 31 (2): 351–367.

Boyle, P., M. Broadie, and P. Glasserman. 1997. "Monte Carlo Methods for Security Pricing". *Journal of Economic Dynamics and Control* 21 (8-9): 1267–1321.

Chen, Z., and P. Glasserman. 2008. "Fast Pricing of Basket Default Swaps". *Operations Research* 56 (2): 286–303.

Crouhy, M., D. Galai, and R. Mark. 2000. "A Comparative Analysis of Current Credit Risk Models". *Journal of Banking & Finance* 24 (1-2): 59–117.

Fienberg, S. 1974. "Stochastic Models for Single Neuron Firing Trains: a Survey". *Biometrics* 30 (3): 399–427.

Gerstein, G., and B. Mandelbrot. 1964. "Random Walk Models for the Spike Activity of a Single Neuron". *Biophysical Journal* 4 (1): 41–68.

Giles, M. 2008. "Multilevel Monte Carlo Path Simulation". *Operations Research* 56 (3): 607–617.

Glasserman, P. 2004. *Monte Carlo Methods in Financial Engineering*. Springer Verlag.

Gobet, E., and S. Menozzi. 2004. "Exact Approximation Rate of Killed Hypoelliptic Diffusions Using the Discrete Euler Scheme". *Stochastic Processes and their Applications* 112 (2): 201–223.

Gobet, E., and S. Menozzi. 2007. "Discrete Sampling of Functionals of Itô Processes". *Séminaire de Probabilités XL* XL:355–374.

Gobet, E., and S. Menozzi. 2010. "Stopped Diffusion Processes: Boundary Corrections and Overshoot". *Stochastic Processes and their Applications* 120 (2): 130–162.

Heinrich, S. 2001. "Multilevel Monte Carlo Methods". In *Lecture Notes in Large Scale Scientific Computing*, edited by S. Margenov, J. Wasniewski, and P. Yalamov, 58–67: Springer-Verlag.

Higham, D., X. Mao, M. Roj, Q. Song, and G. Yin. 2013. "Mean Exit Times and the Multi-level Monte Carlo Method". *SIAM Journal on Uncertainty Quantification* to appear.

Iyengar, S. 1985. "Hitting Lines with Two-dimensional Brownian Motion". *SIAM Journal on Applied Mathematics* 45:983–989.

Joro, T., A. Niu, and P. Na. 2004, December. "A Simulation-Based First-to-Default (FtD) Credit Default Swap (CDS) Pricing Approach under Jump-Diffusion". In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, Volume 2, 1632–1636. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Longstaff, F., and E. Schwartz. 2001. "Valuing American Options by Simulation: A Simple Least-squares Approach". *Review of Financial Studies* 14 (1): 113.

Mannella, R. 1999. "Absorbing Boundaries and Optimal Stopping in a Stochastic Differential Equation". *Physics Letters A* 254 (5): 257–262.

AUTHOR BIOGRAPHIES

DESMOND J. HIGHAM is a professor and 1966 Chair of Numerical Analysis in the Department of Mathematics and Statistics at the University of Strathclyde. He is a Fellow of the Royal Society of Edinburgh and a Fellow of the Society for Industrial and Applied Mathematics. He serves as Section Editor of the Survey and Review section of Society for Industrial and Applied Mathematics (SIAM) Review and is on the editorial board of a number of journals, such as Proceedings of the Royal Society A, IMA Journal of

Numerical Analysis and Journal of Computational Finance. His research interests lie broadly in numerical analysis, stochastic computation and complexity science. His email address is d.j.higham@strath.ac.uk and his web page is http://personal.strath.ac.uk/d.j.higham.

MIKOLAJ ROJ is a Ph.D. student in the Department of Mathematics and Statistics at the University of Strathclyde. He received an M.Sc. in Financial and Insurance Mathematics from the Technical University of Lodz and an M.Sc. in Finance and Management from the University of Wales in Aberystwyth. His research interests are in multi-level Monte Carlo, risk management, efficient modeling and simulation solutions. His email address is mikolaj.roj@strath.ac.uk and his web page is http://personal.strath.ac.uk/mikolaj.roj.