Multilevel Monte Carlo
path simulation

Mike Giles

giles@comlab.ox.ac.uk

Oxford University Computing Laboratory
Oxford-Man Institute of Quantitative Finance

Acknowledgments: research funding from Microsoft and EPSRC, and collaboration with Paul Glasserman (Columbia) and Ian Sloan, Frances Kuo (UNSW)
Outline

Long-term objective is faster Monte Carlo simulation of path dependent options to estimate values and Greeks.

Several ingredients, not yet all combined:
- multilevel method
- quasi-Monte Carlo
- adjoint pathwise Greeks
- parallel computing on NVIDIA graphics cards

Emphasis in this presentation is on multilevel method
Generic Problem

Stochastic differential equation with general drift and volatility terms:

\[ dS(t) = a(S, t) \, dt + b(S, t) \, dW(t) \]

We want to compute the expected value of an option dependent on \( S(t) \). In the simplest case of European options, it is a function of the terminal state

\[ P = f(S(T)) \]

with a uniform Lipschitz bound,

\[ |f(U) - f(V)| \leq c \, \|U - V\|, \quad \forall U, V. \]
Simplest MC Approach

Euler discretisation with timestep $h$:

$$\hat{S}_{n+1} = \hat{S}_n + a(\hat{S}_n, t_n) h + b(\hat{S}_n, t_n) \Delta W_n$$

Estimator for expected payoff is an average of $N$ independent path simulations:

$$\hat{Y} = N^{-1} \sum_{i=1}^{N} f(\hat{S}^{(i)}_{T/h})$$

- weak convergence – $O(h)$ error in expected payoff
- strong convergence – $O(h^{1/2})$ error in individual path
Simplest MC Approach

Mean Square Error is $O \left( N^{-1} + h^2 \right)$

- first term comes from variance of estimator
- second term comes from bias due to weak convergence

To make this $O(\varepsilon^2)$ requires

$$N = O(\varepsilon^{-2}), \quad h = O(\varepsilon) \quad \Rightarrow \quad \text{cost} = O(N h^{-1}) = O(\varepsilon^{-3})$$

Aim is to improve this cost to $O \left( \varepsilon^{-p} \right)$, with $p$ as small as possible, ideally close to 1.

Note: for a relative error of $\varepsilon = 0.001$, the difference between $\varepsilon^{-3}$ and $\varepsilon^{-1}$ is huge.
Standard MC Improvements

- variance reduction techniques (e.g. control variates, stratified sampling) improve the constant factor in front of $\varepsilon^{-3}$, sometimes spectacularly

- improved second order weak convergence (e.g. through Richardson extrapolation) leads to $h = O(\sqrt{\varepsilon})$, giving $p = 2.5$

- quasi-Monte Carlo reduces the number of samples required, at best leading to $N \approx O(\varepsilon^{-1})$, giving $p \approx 2$ with first order weak methods

Multilevel method gives $p = 2$ without QMC, and at best $p \approx 1$ with QMC.
Other Related Research

In Dec. 2005, Ahmed Kebaier published an article in *Annals of Applied Probability* describing a two-level method which reduces the cost to $O(\varepsilon^{-2.5})$.

Also in Dec. 2005, Adam Speight wrote a working paper describing a very similar multilevel use of control variates.

There are also close similarities to a multilevel technique developed by Stefan Heinrich for parametric integration (*Journal of Complexity*, 1998).
Multilevel MC Approach

Consider multiple sets of simulations with different timesteps $h_l = 2^{-l}T$, $l = 0, 1, \ldots, L$, and payoff $\hat{P}_l$

$$\mathbb{E}[\hat{P}_L] = \mathbb{E}[\hat{P}_0] + \sum_{l=1}^{L} \mathbb{E}[\hat{P}_l - \hat{P}_{l-1}]$$

Expected value is same – aim is to reduce variance of estimator for a fixed computational cost.

Key point: approximate $\mathbb{E}[\hat{P}_l - \hat{P}_{l-1}]$ using $N_l$ simulations with $\hat{P}_l$ and $\hat{P}_{l-1}$ obtained using same Brownian path.

$$\hat{Y}_l = N_l^{-1} \sum_{i=1}^{N_l} \left( \hat{P}_l^{(i)} - \hat{P}_{l-1}^{(i)} \right)$$
Multilevel MC Approach

Discrete Brownian path at different levels
Multilevel MC Approach

Using independent paths for each level, the variance of the combined estimator is

$$\mathbb{V} \left[ \sum_{l=0}^{L} \hat{Y}_l \right] = \sum_{l=0}^{L} N_l^{-1} V_l, \quad V_l \equiv \mathbb{V} [\hat{P}_l - \hat{P}_{l-1}] ,$$

and the computational cost is proportional to $\sum_{l=0}^{L} N_l h_l^{-1}$.

Hence, the variance is minimised for a fixed computational cost by choosing $N_l$ to be proportional to $\sqrt{V_l h_l}$.

The constant of proportionality can be chosen so that the combined variance is $O(\varepsilon^2)$. 
Multilevel MC Approach

For the Euler discretisation and a Lipschitz payoff function

$$\nabla[\hat{P}_l - P] = O(h_l) \iff \nabla[\hat{P}_l - \hat{P}_{l-1}] = O(h_l)$$

and the optimal $N_l$ is asymptotically proportional to $h_l$. To make the combined variance $O(\varepsilon^2)$ requires

$$N_l = O(\varepsilon^{-2} L h_l).$$

To make the bias $O(\varepsilon)$ requires

$$L = \log_2 \varepsilon^{-1} + O(1) \implies h_L = O(\varepsilon).$$

Hence, we obtain an $O(\varepsilon^2)$ MSE for a computational cost which is $O(\varepsilon^{-2} L^2) = O(\varepsilon^{-2}(\log \varepsilon)^2)$. 

Multilevel Monte Carlo – p. 11/41
Theorem: Let $P$ be a functional of the solution of a stochastic o.d.e., and $\hat{P}_l$ the discrete approximation using a timestep $h_l = M^{-l} T$.

If there exist independent estimators $\hat{Y}_l$ based on $N_l$ Monte Carlo samples, and positive constants $\alpha \geq \frac{1}{2}$, $\beta$, $c_1$, $c_2$, $c_3$ such that

i) $\mathbb{E}[\hat{P}_l - P] \leq c_1 h_l^\alpha$

ii) $\mathbb{E}[\hat{Y}_l] = \begin{cases} \mathbb{E}[\hat{P}_0], & l = 0 \\ \mathbb{E}[\hat{P}_l - \hat{P}_{l-1}], & l > 0 \end{cases}$

iii) $\mathbb{V}[\hat{Y}_l] \leq c_2 N_l^{-1} h_l^\beta$

iv) $C_l$, the computational complexity of $\hat{Y}_l$, is bounded by

$$C_l \leq c_3 N_l h_l^{-1}$$
then there exists a positive constant $c_4$ such that for any $\varepsilon < e^{-1}$ there are values $L$ and $N_l$ for which the multi-level estimator

$$\hat{Y} = \sum_{l=0}^{L} \hat{Y}_l,$$

has Mean Square Error $MSE \equiv \mathbb{E} \left[ \left( \hat{Y} - \mathbb{E}[P] \right)^2 \right] < \varepsilon^2$

with a computational complexity $C$ with bound

$$C \leq \begin{cases} 
  c_4 \varepsilon^{-2}, & \beta > 1, \\
  c_4 \varepsilon^{-2} (\log \varepsilon)^2, & \beta = 1, \\
  c_4 \varepsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1.
\end{cases}$$
The theorem suggests use of Milstein scheme — better strong convergence, same weak convergence

Generic scalar SDE:

\[ dS(t) = a(S, t) \, dt + b(S, t) \, dW(t), \quad 0 < t < T. \]

Milstein scheme:

\[ \hat{S}_{n+1} = \hat{S}_n + a \, h + b \, \Delta W_n + \frac{1}{2} b' b \left( (\Delta W_n)^2 - h \right). \]
Milstein Scheme

In scalar case:

- $O(h)$ strong convergence
- $O(\varepsilon^{-2})$ complexity for Lipschitz payoffs – trivial
- $O(\varepsilon^{-2})$ complexity for Asian, lookback, barrier and digital options using carefully constructed estimators based on Brownian interpolation or extrapolation
Milstein Scheme

Key idea: within each timestep, model the behaviour as simple Brownian motion conditional on the two end-points

\[ \hat{S}(t) = \hat{S}_n + \lambda(t)(\hat{S}_{n+1} - \hat{S}_n) \]
\[ + b_n \left( W(t) - W_n - \lambda(t)(W_{n+1} - W_n) \right), \]

where

\[ \lambda(t) = \frac{t - t_n}{t_{n+1} - t_n} \]

There then exist analytic results for the distribution of the min/max/average over each timstep.
Results

Geometric Brownian motion:

\[ dS = r S \, dt + \sigma S \, dW, \quad 0 < t < T, \]

with parameters \( T = 1, \quad S(0) = 1, \quad r = 0.05, \quad \sigma = 0.2 \)

- **European call option:** \( \exp(-rT) \max(S(T) - 1, 0) \)
- **Asian option:** \( \exp(-rT) \max \left( T^{-1} \int_0^T S(t) \, dt - 1, 0 \right) \)
- **Lookback option:** \( \exp(-rT) \left( S(T) - \min_{0<t<T} S(t) \right) \)
- **Down-and-out barrier option:** same as call provided \( S(t) \) stays above \( B = 0.9 \)
MLMC Results

GBM: European

\[ \log_2 \text{variance} \]

\[ \log_2 |\text{mean}| \]

call
GBM: European

\[ \varepsilon = 0.00005 \quad \varepsilon = 0.0001 \quad \varepsilon = 0.0002 \quad \varepsilon = 0.0005 \quad \varepsilon = 0.001 \]

\[ N_i \]

\[ \varepsilon^2 \text{ Cost} \]

\[ \varepsilon \]

Multilevel Monte Carlo – p. 19/41
MLMC Results

GBM: Asian

![Graphs showing log variance and log mean for option pricing with Multilevel Monte Carlo (MLMC)]
MLMC Results

GBM: Asian

![Graph showing the convergence of the MLMC method for Asian options with different tolerances. The x-axis represents the level of refinement, and the y-axis represents the number of samples. The plot compares the cost of standard Monte Carlo (Std MC) and MLMC methods.]
MLMC Results

GBM: lookback

\[ \log_2 \text{variance} \]

\[ \log_2 |\text{mean}| \]

option

Multilevel Monte Carlo – p. 22/41
MLMC Results

GBM: lookback

![Graph showing the relationship between $N_l$ and $\varepsilon$ for different levels of option precision.]

![Graph showing the relationship between $\varepsilon^2$ Cost and $\varepsilon$.]

Multilevel Monte Carlo – p. 23/41
MLMC Results

GBM: barrier

option

Multilevel Monte Carlo – p. 24/41
MLMC Results

GBM: barrier

option
Milstein Scheme

Generic vector SDE:

\[ dS(t) = a(S, t) \, dt + b(S, t) \, dW(t), \quad 0 < t < T, \]

with correlation matrix \( \Omega(S, t) \) between elements of \( dW(t) \).

Milstein scheme:

\[ \hat{S}_{i,n+1} = \hat{S}_{i,n} + a_i \, h + b_{ij} \Delta W_{j,n} \]

\[ + \frac{1}{2} \frac{\partial b_{ij}}{\partial S_l} b_{lk} \left( \Delta W_{j,n} \Delta W_{k,n} - h \Omega_{jk} - A_{jk,n} \right) \]

with implied summation, and Lévy areas defined as

\[ A_{jk,n} = \int_{t_n}^{t_{n+1}} (W_j(t) - W_j(t_n)) \, dW_k - (W_k(t) - W_k(t_n)) \, dW_j. \]
Milstein Scheme

In vector case:

- $O(h)$ strong convergence if Lévy areas are simulated correctly – expensive

- $O(h^{1/2})$ strong convergence in general if Lévy areas are omitted, except if a certain commutativity condition is satisfied (useful for a number of real cases)

- Lipschitz payoffs can be handled well using antithetic variables

- Other cases may require approximate simulation of Lévy areas
Results

Heston model:

\[ \begin{align*}
    \text{d}S &= r\, S\, \text{d}t + \sqrt{V} \, S \, \text{d}W_1, \quad 0 < t < T \\
    \text{d}V &= \lambda (\sigma^2 - V) \, \text{d}t + \xi \sqrt{V} \, \text{d}W_2, \\
    T &= 1, \quad S(0) = 1, \quad V(0) = 0.04, \quad r = 0.05, \\
    \sigma &= 0.2, \quad \lambda = 5, \quad \xi = 0.25, \quad \rho = -0.5
\end{align*} \]
MLMC Results

Heston model: European call

\[
\log_2 \text{variance} \quad P_l \quad P_l - P_{l-1}
\]

\[
\log_2 |\text{mean}| \quad P_l \quad P_l - P_{l-1}
\]
MLMC Results

Heston model: European call

![Graphs showing MLMC results for different ε values](image)

- **N⁻** vs. **l**
- **ε² Cost** vs. **ε**

- **ε=0.00005**
- **ε=0.0001**
- **ε=0.0002**
- **ε=0.0005**
- **ε=0.001**

Multilevel Monte Carlo – p. 30/41
Quasi-Monte Carlo

- well-established technique for approximating high-dimensional integrals
- for finance applications see papers by l’Ecuyer and book by Glasserman
- Sobol sequences are perhaps most popular; we use lattice rules (Sloan & Kuo)
- two important ingredients for success:
  - randomized QMC for confidence intervals
  - good identification of “dominant dimensions” (Brownian Bridge and/or PCA)
Quasi-Monte Carlo

Approximate high-dimensional hypercube integral

$$\int_{[0,1]^d} f(x) \, dx$$

by

$$\frac{1}{N} \sum_{i=0}^{N-1} f(x^{(i)})$$

where

$$x^{(i)} = \left[ \frac{i}{N} z \right]$$

and $z$ is a $d$-dimensional “generating vector”.

Multilevel Monte Carlo – p. 32/41
Quasi-Monte Carlo

In the best cases, error is $O(N^{-1})$ instead of $O(N^{-1/2})$ but without a confidence interval.

To get a confidence interval, let

$$x(i) = \left[ \frac{i}{N} \z + x_0 \right].$$

where $x_0$ is a random offset vector.

Using 32 different random offsets gives a confidence interval in the usual way.
Quasi-Monte Carlo

For the path discretisation we can use

\[ \Delta W_n = \sqrt{h} \Phi^{-1}(x_n), \]

where \( \Phi^{-1} \) is the inverse cumulative Normal distribution.

Much better to use a Brownian Bridge construction:

- \( x_1 \)  \( \longrightarrow \)  \( W(T) \)
- \( x_2 \)  \( \longrightarrow \)  \( W(T/2) \)
- \( x_3, x_4 \)  \( \longrightarrow \)  \( W(T/4), W(3T/4) \)
- \( \ldots \) and so on by recursive bisection
Multilevel QMC

- rank-1 lattice rule developed by Sloan, Kuo & Waterhouse at UNSW
- 32 randomly-shifted sets of QMC points
- number of points in each set increased as needed to achieved desired accuracy, based on confidence interval estimate
- results show QMC to be particularly effective on lowest levels with low dimensionality
MLQMC Results

GBM: European call

![Graph showing log2 variance and log2 mean for different levels of variance and mean with different markers and line styles.]

Multilevel Monte Carlo – p. 36/41
MLQMC Results

GBM: European call

![Graph showing GBM European call results with different error rates and cost curves.](image)
MLQMC Results

GBM: barrier option

![Graphs showing log2 variance and log2 |mean| for different levels of refinement (1, 16, 256, 4096) and the difference between P_l and P_{l-1}.]
MLQMC Results

GBM: barrier option

![Diagram showing results of MLQMC in GBM barrier option](image)

- ε = 0.00005
- ε = 0.0001
- ε = 0.0002
- ε = 0.0005
- ε = 0.001

ε² Cost vs. ε

MLQMC vs. Std QMC
Conclusions

Results so far:

- much improved order of complexity
- fairly easy to implement
- significant benefits for model problems

However:

- lots of scope for further development
  - multi-dimensional SDEs needing Lévy areas
  - adjoint Greeks and “vibrato” Monte Carlo
  - numerical analysis of algorithms
  - execution on NVIDIA graphics cards (128 cores)
- need to test ideas on real finance applications
Papers


www.comlab.ox.ac.uk/mike.giles/finance.html

Email: giles@comlab.ox.ac.uk