Multi-level Monte Carlo in Stochastic Simulation

Des Higham
Department of Mathematics and Statistics
University of Strathclyde

Outline

- Weak versus strong convergence
- Complexity of Monte Carlo
- Multi-level Monte Carlo
- Financial Options
- Hitting Times
- Gillespie/Tau leaping

Multi-level Monte Carlo

Heinrich, *Lect. Notes Comput. Sci.*, **2001**
Giles, *Operations Research*, **2008** (78 citatons)

Path-dependent expectations

Giles, Higham, Mao, *Finance and Stoch.*, **2009**

Mean exit times

Higham, Mao, Roj, Song, *Tech. Report*, **2011**

Gillespie/Tau leaping

Anderson, Higham, *submitted*

Weak versus Strong

SDE:

$$d\mathbf{S}(t) = a(\mathbf{S}(t))\,dt + b(\mathbf{S}(t))\,d\mathbf{W}(t)$$

$\mathbf{S}(0)$ given and $0 \le t \le T$

Euler–Maruyama

$$\mathbf{S}_{n+1} = \mathbf{S}_n + a(\mathbf{S}_n)h + b(\mathbf{S}_n)\Delta\mathbf{W}_n$$

$\Delta\mathbf{W}_n := \mathbf{W}(t_{n+1}) - \mathbf{W}(t_n), \quad t_n = nh, \quad h = T/K$

Assume that a and b are smooth and globally Lipschitz

Weak versus Strong

Weak Convergence $\left|\mathbb{E}\left[\mathbf{S}(t_n)\right] - \mathbb{E}\left[\mathbf{S}_n\right]\right| \leq Ch$

Strong Convergence

$$\mathbb{E}\left[\sup_{0\leq n\leq K} \left|\mathbf{S}(t_n) - \mathbf{S}_n\right|\right] \leq Ch^{\frac{1}{2}}$$

Strong convergence $+$ Markov inequality $\Rightarrow$

$$\mathbf{P}\left(\left|\mathbf{S}(t_n) - \mathbf{S}_n\right| \geq h^{\alpha}\right) \leq Ch^{\frac{1}{2}\ -\ \alpha}$$

Continuous Time/Higher Moments

$$\mathbb{E}\left[\sup_{0\leq t\leq T} \left|\mathbf{S}(t) - \mathbf{S}(t)\right|^{m}\right] \leq C_{m,\delta} h^{\frac{m}{2}-\delta}$$

Weak versus Strong

Which is more relevant, **weak** or **strong**?

Conventional wisdom :

Weak convergence is usually enough. Most problems require **expected value** type information.

Strong convergence covers cases where we want to **visualize paths** or generate **time series** (e.g. to test a filtering algorithm or a parameter fitting algorithm).

Monte Carlo for SDEs

Approximate $\mathbb{E}[\mathbf{S}(T)]$ by applying E-M to get samples.
Let $\mu = \frac{1}{N}\sum_{i=1}^{N} S_K^{[i]}$
Then

$$\begin{aligned}\mathbb{E}[\mathbf{S}(T)] - \mu &= \mathbb{E}[\mathbf{S}(T) - \mathbf{S}_K + \mathbf{S}_K] - \mu \\ &= \mathbb{E}[\mathbf{S}(T) - \mathbf{S}_K] + \mathbb{E}[\mathbf{S}_K] - \mu\end{aligned}$$

Confidence interval width is $O(h) + O(1/\sqrt{N})$

For confidence interval of $O(\epsilon)$, choose $h = 1/\sqrt{N} = \epsilon$

Computational cost is $N \times 1/h$

Hence, computational complexity is $O(\epsilon^{-3})$

Multi-level Monte Carlo

The **Multi-level Monte Carlo** algorithm will achieve computational complexity of

$$O(\epsilon^{-2}\log(\epsilon)^2)$$

using E-M, and giving good results in practice

A key ingredient: Use a range of h values
many paths at large h, few paths at small h

Multi-level Monte Carlo

Consider payoff $f(\mathbf{S}(T))$, where f is globally Lipschitz.
ϵ is required accuracy (conf. int.)

Timesteps $h_l = M^{-l}T, \quad l = 0, 1, 2, \ldots, L$

M is fixed and $L = \frac{\log \epsilon^{-1}}{\log M}$, so that $h_L = O(\epsilon)$

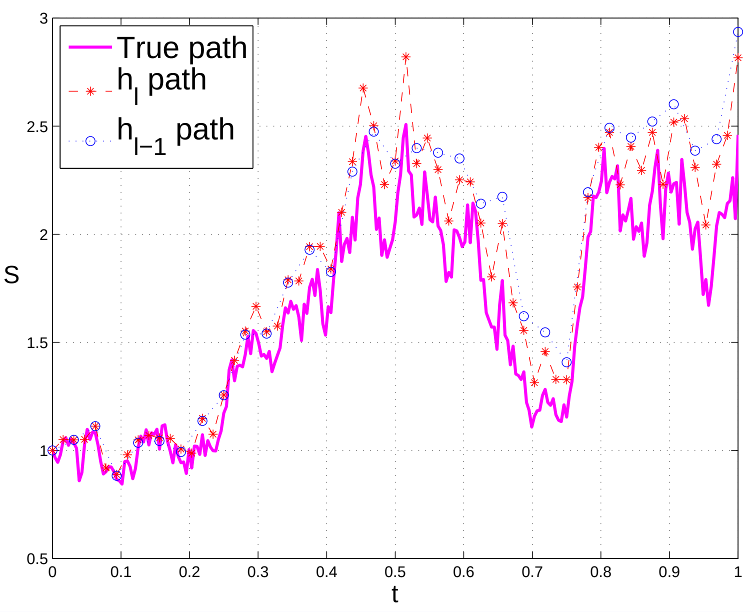
$\widehat{\mathbf{P}}_l$ denotes E-M approx. to $f(\mathbf{S}(T))$ using h_l. Clearly

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

$\widehat{Y}_0$ estimates $\mathbb{E}[\widehat{\mathbf{P}}_0]$ using N_0 paths, and
$\widehat{Y}_l$ estimates $\mathbb{E}[\widehat{\mathbf{P}}_l - \widehat{\mathbf{P}}_{l-1}]$ using N_l paths:

$$\widehat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\widehat{P}_l^{[i]} - \widehat{P}_{l-1}^{[i]} \right)$$

Multi-level Monte Carlo ($M = 2$)



Multi-level Monte Carlo

Strong convergence of E-M $+$ glob. Lip. f give

$$\mathrm{var}\left[\widehat{\mathbf{P}}_l - f\left(\mathbf{S}(T)\right)\right] \le \mathbb{E}\left[\left(\widehat{\mathbf{P}}_l - f\left(\mathbf{S}(T)\right)\right)^2\right] = O(h_l)$$

and

$$\mathrm{var}\left[\widehat{\mathbf{P}}_l - \widehat{\mathbf{P}}_{l-1}\right]$$

$$\le \left(\sqrt{\mathrm{var}\left[\widehat{\mathbf{P}}_l - f\left(\mathbf{S}(T)\right)\right]} + \sqrt{\mathrm{var}\left[\widehat{\mathbf{P}}_{l-1} - f\left(\mathbf{S}(T)\right)\right]}\right)^2 = O(h_l)$$

So $\widehat{Y}_l = \frac{1}{N_l}\sum_{i=1}^{N_l}\left(\widehat{P}_l^{[i]} - \widehat{P}_{l-1}^{[i]}\right)$ has variance of $O(h_l/N_l)$

Recap: $\mathbb{E}\left[\mathbf{P}_L\right] = \mathbb{E}\left[\mathbf{P}_0\right] + \sum_{l=1}^{L} \mathbb{E}\left[\mathbf{P}_l - \mathbf{P}_{l-1}\right]$

Estimator for RHS is $\widehat{Y} := \widehat{Y}_0 + \sum_{l=1}^{L} \widehat{Y}_l$

For $l > 1$, $\widehat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\widehat{P}_l^{[i]} - \widehat{P}_{l-1}^{[i]}\right)$ and

$\mathrm{var}\left[\widehat{Y}_l\right] = O(h_l/N_l) \Rightarrow \mathrm{var}\left[\widehat{Y}\right] = \mathrm{var}\left[\widehat{Y}_0\right] + \sum_{l=1}^{L} O(h_l/N_l)$

Take $N_l = O(\epsilon^{-2} L h_l)$, to give $\mathrm{var}\left[\widehat{Y}\right] = O(\epsilon^2)$

Because $h_L = O(\epsilon)$, the bias $\mathbb{E}\left[\widehat{\mathbf{P}}_L - f(\mathbf{S}(T))\right] = O(\epsilon)$

Computational complexity is

$$\sum_{l=0}^{L} N_l h_l^{-1} = \sum_{l=0}^{L} \epsilon^{-2} L h_l h_l^{-1} = L^2 \epsilon^{-2}$$

Since $L = \frac{\log \epsilon^{-1}}{\log M}$, this gives $O(\epsilon^{-2} (\log \epsilon)^2)$

Now $\mathbf{S}(t)$ represents the **asset price**

Option Payoffs :

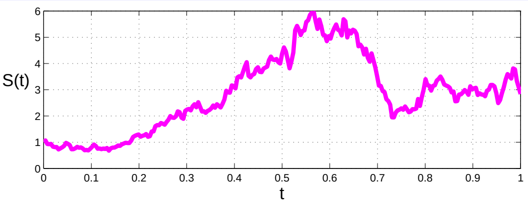
European call: $\max\left(\mathbf{S}(T) - E, 0\right)$

Digital: $\mathbf{1}_{\mathbf{S}(T)>E}$

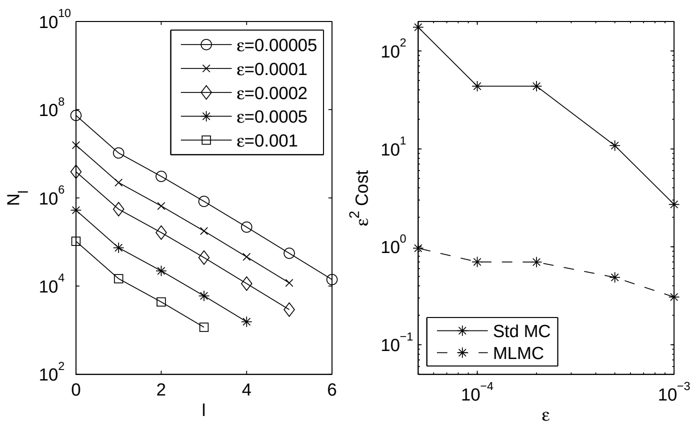
Lookback: $\mathbf{S}(T) - \min_{0\le t\le T} \mathbf{S}(t)$

Up and out: $\max\left(\mathbf{S}(T) - E, 0\right) \times \mathbf{1}_{\left(\sup_{0\le t\le T} \mathbf{S}(t)\right)\le B}$

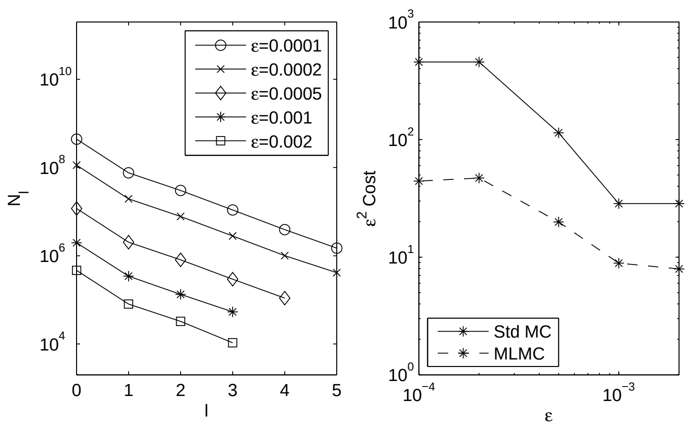
Task: compute $\mathbb{E}\left[\text{Payoff}\right]$



Lookback with geom. Brownian motion



Digital with geom. Brownian motion



Payoff Not Globally Lipschitz?

Extending Giles (2008) reduces to getting

$$\mathbb{E}\left[\left(\mathbf{P}-\widehat{\mathbf{P}}\right)^2\right] \leq O\left(h^\beta\right)$$

where
$\mathbf{P}$ is true payoff,
$\widehat{\mathbf{P}}$ is Euler–Maruyama payoff

In Giles, Higham, Mao (2009), we confirmed rigorously that, given any $\delta > 0$,

- $\beta = 1 - \delta$ for a **lookback**
- $\beta = \frac{1}{2} - \delta$ for a **digital**
- $\beta = \frac{1}{2} - \delta$ for a **barrier**

(Still assume SDE coeffs glob. Lipsch. Up and out fits well!)

Stopped Exit Times

Required in many physical modeling scenarios
Look at scalar case for simplicity

Suppose $\mathbf{S}(0) = x \in (\alpha, \beta)$. For the SDE we define

$$\tau := (\inf\{t > 0 \;:\; \mathbf{S}(t) \notin (\alpha, \beta)\}) \wedge T$$

For the E-M approximation

$$\nu := (\inf\{t > 0 \;:\; \mathbf{S}(t) \notin (\alpha, \beta)\}) \wedge T$$

Assumptions

- Drift and diffusion globally Lipschitz and smooth
- Diffusion strictly positive (uniform ellipticity)

This ensures that $u(x) := \mathbb{E}[\tau]$ is Lipschitz

Weak Error in Mean Hitting Time

Gobet & Menozzi, Stoch. Proc. Appl., 2010:

$$\mathbb{E}[\tau] - \mathbb{E}[\nu] = O(h^{\frac{1}{2}})$$

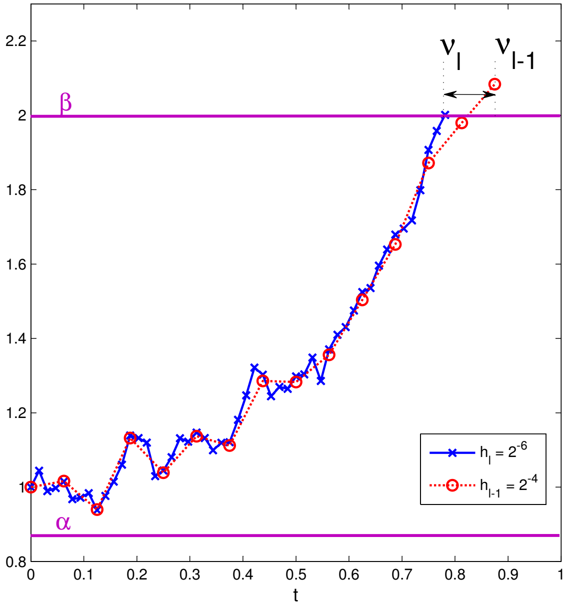
Standard Monte Carlo for accuracy ϵ:
to balance bias and sampling error we need

$$\epsilon = h^{\frac{1}{2}} = 1/\sqrt{N}$$

This gives computational complexity of $O(\epsilon^{-4})$

We will show that multi-level can achieve $O\left(\epsilon^{-3}(\log \epsilon)^2\right)$

Illustration of one sample at one level



Key Result

Strong error in mean exit time

We need to show that

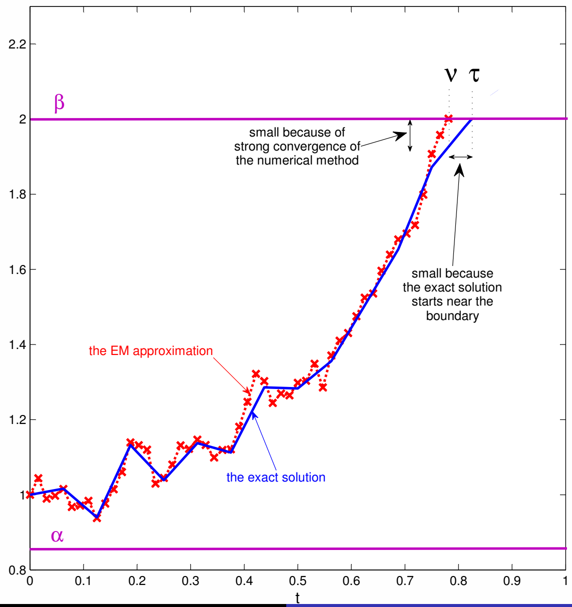
$$\mathbb{E}\left[|\tau - \nu|^2\right] = O(h^{\frac{1}{2}})$$

We use

$$\mathbb{E}\left[|\tau - \nu|^2\right] \leq T\mathbb{E}\left[|\tau - \nu|\right]$$

Then deal separately with the cases $\nu < \tau$ and $\tau < \nu$

Case where $\nu < \tau$



Overall

We can show

$$\mathbb{E}\left[(\tau - \nu)\,\mathbf{1}_{\{\nu<\tau\}}\right] = O(h^{\frac{1}{2}})$$

and

$$\mathbb{E}\left[(\nu - \tau)\,\mathbf{1}_{\{\tau<\nu\}}\right] = O(h^{\frac{1}{2}})$$

So

$$\mathbb{E}\left[|\tau - \nu|\right] = O(h^{\frac{1}{2}})$$

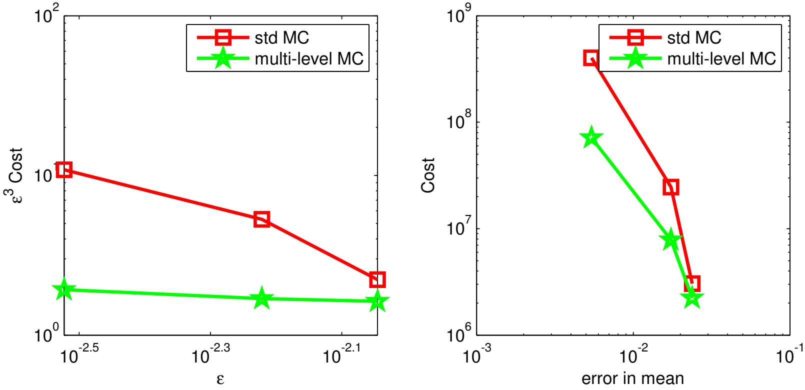
$\Rightarrow$ multi-level version has complexity of

$$O\left(\epsilon^{-3}(\log \epsilon)^2\right)$$

compared to the standard

$$O(\epsilon^{-4})$$

Mean-Reverting Square Root SDE



Gillespie/Tau-leaping

$$
\begin{aligned}
G &\xrightarrow{25} G + M \\
M &\xrightarrow{1000} M + P \\
P + P &\xrightarrow{0.001} D \\
M &\xrightarrow{0.1} \emptyset \\
P &\xrightarrow{1} \emptyset
\end{aligned}
$$

Start with 1 gene
Estimate expected number of dimers at $t = 1$

Method	Solution	Updates	CPU time
Gillespie/MC	3714.6 ± 1	8.3×10^{10}	1.5×10^{5} sec
Tau-leap/MC	3708.4 ± 1	1.7×10^{10}	2.0×10^{4} sec
Tau/Gill/MLMC	3713.9 ± 1	5.8×10^{8}	1.7×10^{3} sec

[Joint work with David Anderson]

Summary

- Multi-level approach dramatically improves Monte Carlo simulation when samples contain discretization errors
- Compute many (cheap) samples at low resolution and few (expensive) samples at high resolution

- Original SDE analysis of Giles (2008) extends to some $\mathbb{E}\left[f(\mathbf{S}(t))\right]$ where f is not globally Lipschitz
- Works for mean exit times
- Now available for Gillespie/tau-leaping

MLMC is currently being pursued in many directions