Multi-Level Monte Carlo Simulations of Mean Exit Times

Mikolaj Roj
Department of Mathematics and Statistics
University of Strathclyde

8th IMACS Seminar on Monte Carlo Methods
Borovets, August 29 – September 2, 2011

EPSRC

nag®

University of Strathclyde

Outline

- Stopped exit times
- Weak versus strong convergence
- Complexity of Monte Carlo

- Multi-level Monte Carlo
  Giles, *Operations Research*, 2008
  Heinrich, *Springer*, 2001

- Mean exit times
  Higham, Mao, Roj, Song, Yin, 2011 submitted

Introduction

Look at scalar case for simplicity.

SDE:

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

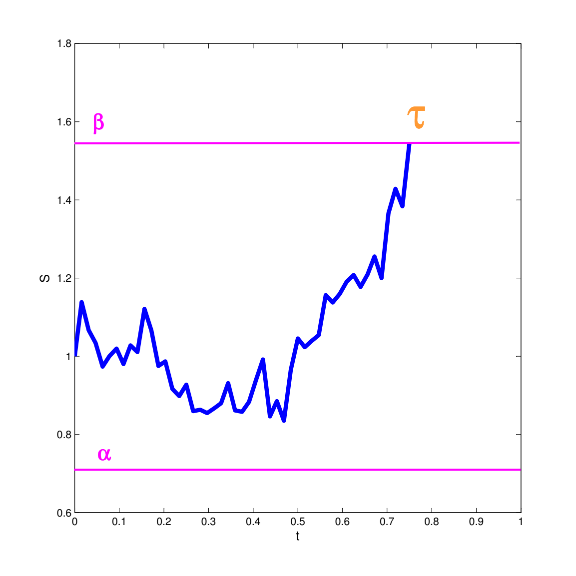
$\mathbf{S}(0)$ given and $0 \leq t \leq T$

Euler–Maruyama

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

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

Stopped Exit Times



Stopped Exit Times

Required in many physical and financial modeling scenarios.

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.

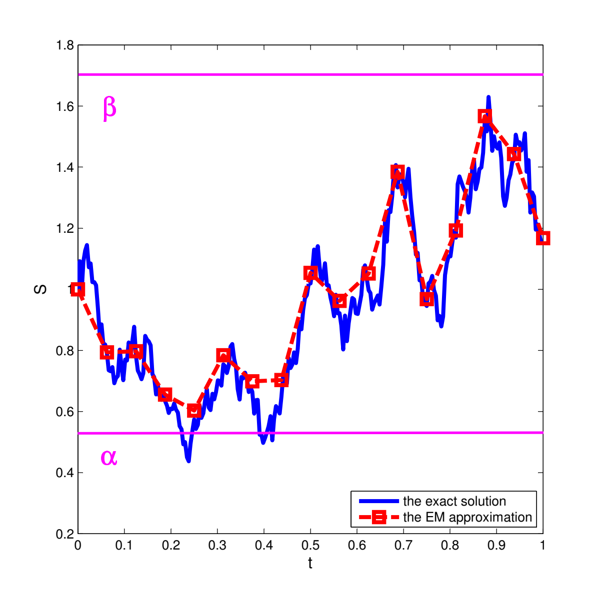
Weak Error in Mean Hitting Time

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

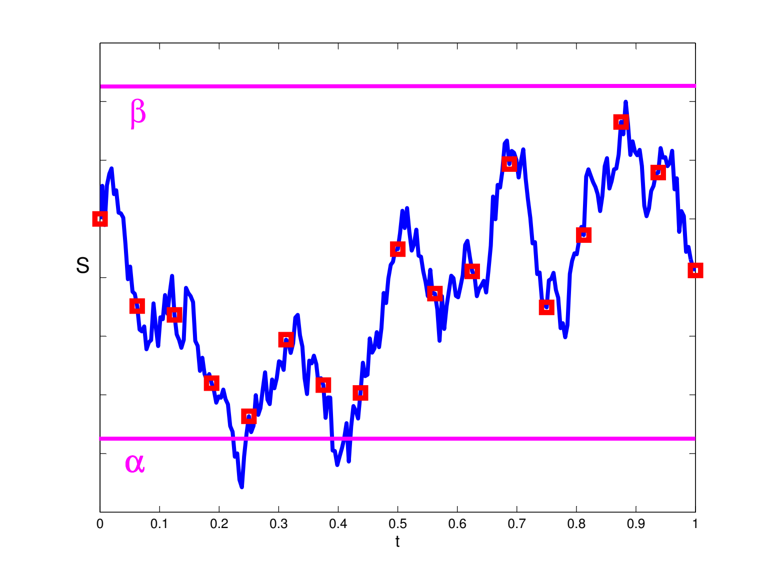
$$\mathbb{E}[\tau] - \mathbb{E}[\nu] = O(h^{\frac{1}{2}})$$

- Large bias in the numerical method
- Exit time samples are less accurate than the corresponding samples of the solution

What Can Go Wrong



Sampling from BM directly



Key Result

Strong Error in Mean Exit Time

We show that

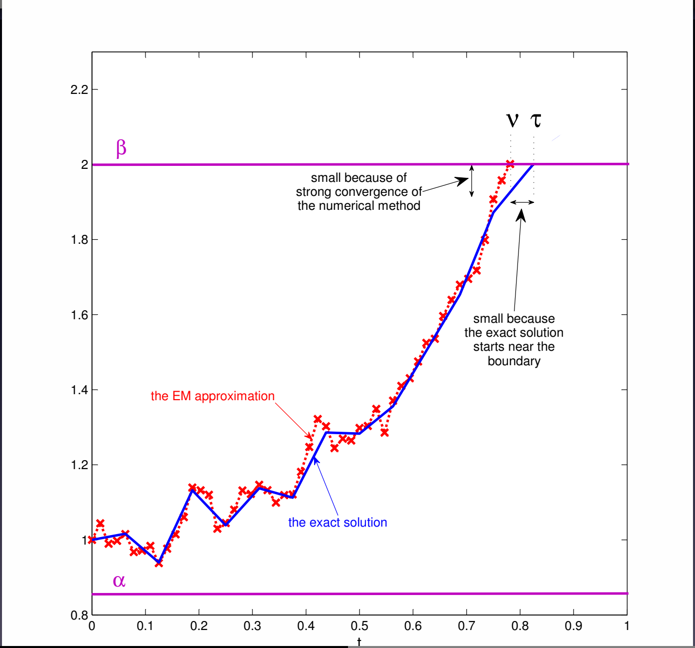
$$\mathbb{E}\left[|\tau - \nu|^p\right] = O(h^{\frac{1}{2}})$$

We use

$$\mathbb{E}\left[|\tau - \nu|^p\right] \leq T^{p-1}\mathbb{E}\left[|\tau - \nu|\right]$$

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

Case I: $\nu < \tau$


2.2
2
1.8
1.6
1.4
1.2
1
0.8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
t
β
α
ν
τ
small because of
strong convergence of
the numerical method
small because
the exact solution
starts near the
boundary
the EM approximation
the exact solution

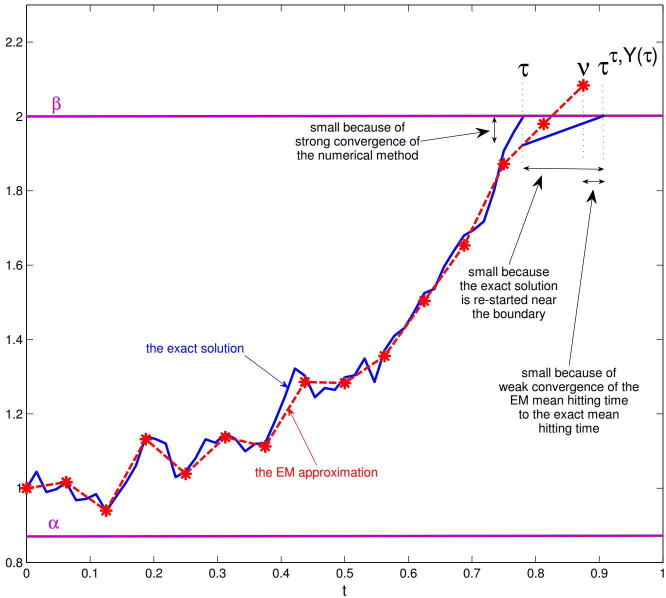
Outline for Case I: $\nu < \tau$

$$\mathbb{E}\left[(\tau - \nu)\,\mathbf{1}_{\{\nu<\tau\}}\right] = \mathbb{E}\left[\mathbb{E}\left[(\tau - \nu)\,\mathbf{1}_{\{\nu<\tau\}}\,|\,\mathcal{F}(\nu)\right]\right]$$

$$= \mathbb{E}\left[\mathbb{E}\left[\tau^{\nu,\mathbf{S}(\nu)}\,|\,\mathcal{F}(\nu)\right]\right]$$

$$\leq K\mathbb{E}\left[\sup_{0\leq u\leq T}|\mathbf{S}(u) - \mathbf{S}(u)|\right] = O(h^{\frac{1}{2}})$$

Case II: $\tau < \nu$



Overall

We have

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

and

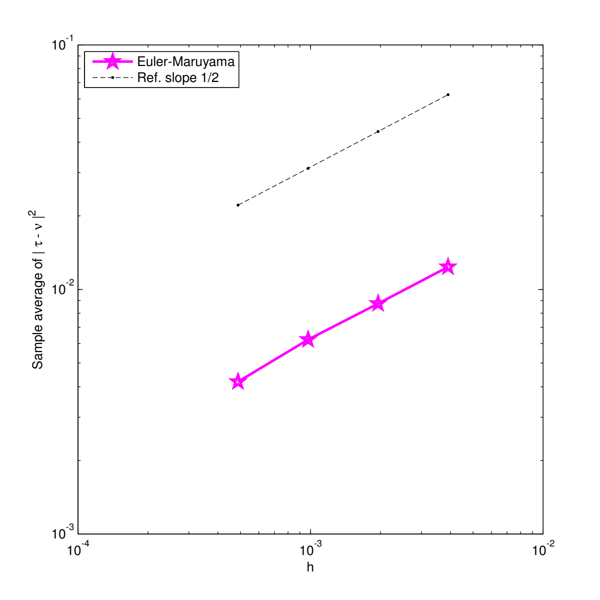
$$\mathbb{E}\left[(\nu - \tau)\,\mathbf{1}_{\{\tau<\nu\}}\right] = O(h^{\frac{1}{2}})$$

So

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

as required

Strong Error in Mean Exit Time



Monte Carlo

Approximate $\mathbb{E}[\tau]$ by applying E-M to get samples

Let $\mu = \frac{1}{N}\sum_{i=1}^{N} \nu^{[i]}$

Then

$$\begin{aligned}
\mathbb{E}[\tau] - \mu &= \mathbb{E}[\tau - \nu + \nu] - \mu \\
&= \mathbb{E}[\tau - \nu] + \mathbb{E}[\nu] - \mu
\end{aligned}$$

Monte Carlo

Approximate $\mathbb{E}[\tau]$ by applying E-M to get samples

Let $\mu = \frac{1}{N}\sum_{i=1}^{N} \nu^{[i]}$

Then

$$\begin{aligned}\mathbb{E}[\tau] - \mu &= \mathbb{E}[\tau - \nu + \nu] - \mu \\ &= \mathbb{E}[\tau - \nu] + \mathbb{E}[\nu] - \mu\end{aligned}$$

Confidence interval width is $O(h^{1/2}) + O(1/\sqrt{N})$

For confidence interval of $O(\epsilon)$, choose $h^{1/2} = 1/\sqrt{N} = \epsilon$

Expected computational cost is $N\times$ expected exit time $\times 1/h$

Hence, computational complexity is $O(\epsilon^{-4})$

Multi-Level Monte Carlo

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

$$O(\epsilon^{-3} \log(\epsilon^{-1}))$$

using E-M, and giving good results in practice

Multi-Level Monte Carlo

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

$$O(\epsilon^{-3} \log(\epsilon^{-1}))$$

using E-M, and giving good results in practice

Key idea: Use a range of h values
many paths at large h, few paths at small h

Multi-level Monte Carlo

ϵ is required accuracy (conf. int.)

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

M is fixed and $L = \frac{\log \epsilon^{-2}}{\log M}$, so that $h_L^{1/2} = O(\epsilon)$

Multi-level Monte Carlo

ϵ is required accuracy (conf. int.)

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

M is fixed and $L = \frac{\log \epsilon^{-2}}{\log M}$, so that $h_L^{1/2} = O(\epsilon)$

$\widehat{\nu}_l$ denotes E-M approx. to τ using h_l. Clearly

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

Multi-level Monte Carlo

ϵ is required accuracy (conf. int.)

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

M is fixed and $L = \frac{\log \epsilon^{-2}}{\log M}$, so that $h_L^{1/2} = O(\epsilon)$

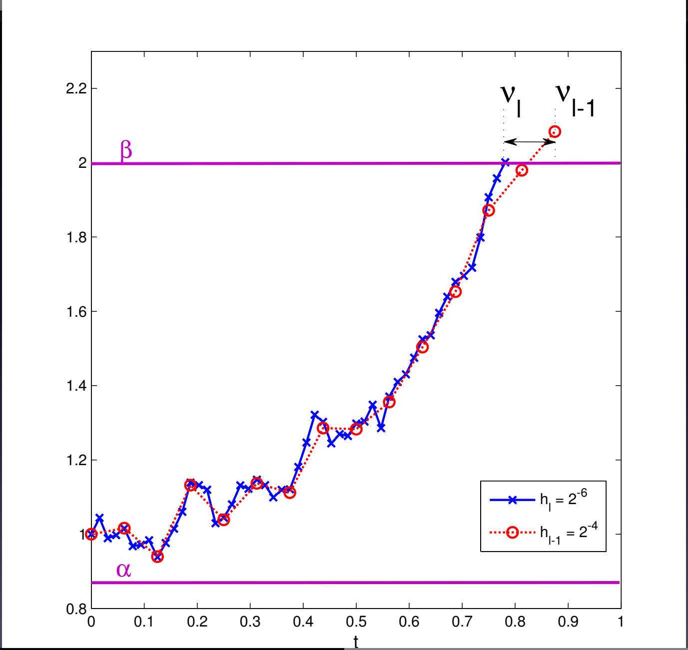
$\widehat{\boldsymbol{\nu}}_l$ denotes E-M approx. to $\boldsymbol{\tau}$ using h_l. Clearly

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

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

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

Multi-level Monte Carlo ($M = 4$)



Multi-level Monte Carlo

Strong convergence of E-M gives

$$\mathrm{var}\left[\widehat{\nu}_l - \tau\right] \leq \mathbb{E}\left[\left(\widehat{\nu}_l - \tau\right)^2\right] = O(h_l^{1/2})$$

Multi-level Monte Carlo

Strong convergence of E-M gives

$$\mathrm{var}\left[\widehat{\nu}_l - \tau\right] \le \mathbb{E}\left[(\widehat{\nu}_l - \tau)^2\right] = O(h_l^{1/2})$$

and

$$\mathrm{var}\left[\widehat{\nu}_l - \widehat{\nu}_{l-1}\right] \le \left(\sqrt{\mathrm{var}\left[\widehat{\nu}_l - \tau\right]} + \sqrt{\mathrm{var}\left[\widehat{\nu}_{l-1} - \tau\right]}\right)^2 = O(h_l^{1/2})$$

So $\widehat{Y}_l = \frac{1}{N_l}\sum_{i=1}^{N_l}\left(\widehat{\nu}_l^{[i]} - \widehat{\nu}_{l-1}^{[i]}\right)$ has variance of $O(h_l^{1/2}/N_l)$

Estimator for RHS is

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

For $l > 1$, $\widehat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\widehat{\nu}_l^{[i]} - \widehat{\nu}_{l-1}^{[i]} \right)$ and var $\left[\widehat{Y}_l \right] = O(h_l^{1/2}/N_l)$

This gives

$$\text{var}\left[\widehat{Y}\right] = \text{var}\left[\widehat{Y}_0\right] + \sum_{l=1}^{L} O(h_l^{1/2}/N_l)$$

Recap: $\mathbb{E}[\widehat{\nu}_L] = \mathbb{E}[\widehat{\nu}_0] + \sum_{l=1}^{L} \mathbb{E}[\widehat{\nu}_l - \widehat{\nu}_{l-1}]$

Estimator for RHS is

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

For $l > 1$, $\widehat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\widehat{\nu}_l^{[i]} - \widehat{\nu}_{l-1}^{[i]} \right)$ and $\text{var}\left[\widehat{Y}_l\right] = O(h_l^{1/2}/N_l)$

This gives

$$\text{var}\left[\widehat{Y}\right] = \text{var}\left[\widehat{Y}_0\right] + \sum_{l=1}^{L} O(h_l^{1/2}/N_l)$$

Take $N_l = O(\epsilon^{-2} L h_l^{1/2})$, to give $\text{var}\left[\widehat{Y}\right] = O(\epsilon^2)$

Because $h_L^{1/2} = O(\epsilon)$, the bias $\mathbb{E}[\widehat{\nu}_L - \tau] = O(\epsilon)$

Expected Computational Cost

Expected computational complexity for MLMC is

$$\sum_{l=0}^{L} N_l h_l^{-1} \mathbb{E}\left[\nu_l\right] = \sum_{l=0}^{L} \epsilon^{-2} L h_l^{1/2} h_l^{-1} \left(\mathbb{E}[\tau] + O(h_l^{1/2})\right)$$

This gives

$$O\left(\epsilon^{-2} L M^{L/2}\right)$$

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

$$O(\epsilon^{-3}(\log \epsilon^{-1}))$$

Std Monte Carlo vs. Multi-Level Monte Carlo

Multi-level version has complexity of

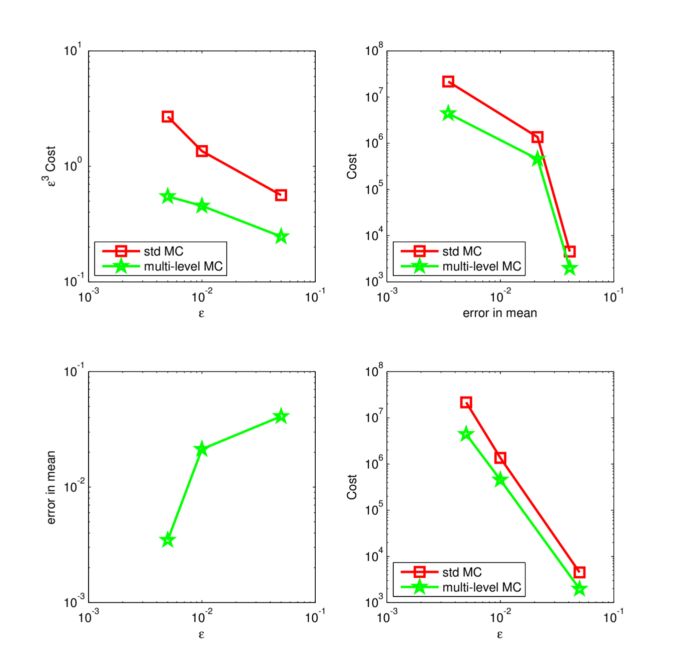
$$O\left(\epsilon^{-3}(\log \epsilon^{-1})\right)$$

compared to the standard

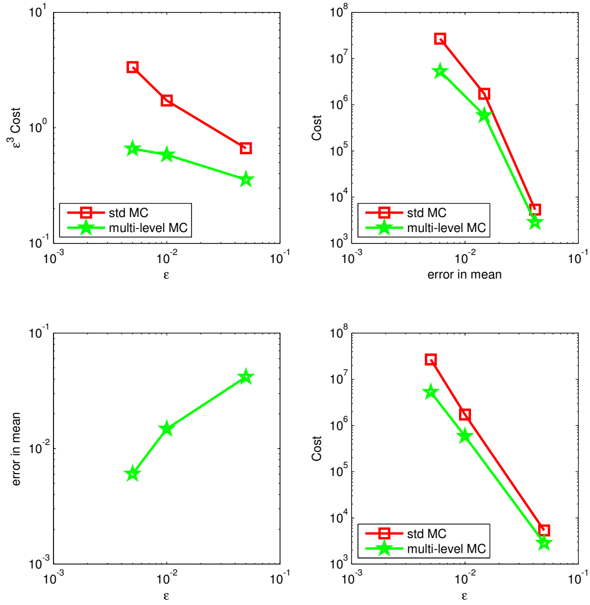
$$O(\epsilon^{-4})$$

which is confirmed computationally too...

Geometric Brownian Motion



Mean Reverting Square Root Process



Summary

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

- The original analysis of Giles (2008) extends to the more computationally challenging task of computing mean exit times

Further work

Further work for mean exit times includes

- Exit time over $[0, \infty)$
- Analysis for higher order discretization methods
- Development of a practical multi-level algorithm