Monte Carlo Methods

Prof. Mike Giles

mike.giles@maths.ox.ac.uk

Oxford University Mathematical Institute

Geometric Brownian Motion

In the case of Geometric Brownian Motion

$$\mathrm{d}S_t = r\,S_t\,\mathrm{d}t + \sigma\,S_t\,\mathrm{d}W_t$$

the use of Itô calculus gives

$$\mathrm{d}(\log S_t) = \left(r - \tfrac{1}{2}\sigma^2\right)\mathrm{d}t + \sigma\,\mathrm{d}W_t$$

which can be integrated to give

$$S_T = S_0\;\exp\left((r - \tfrac{1}{2}\sigma^2)T + \sigma\,W_T\right)$$

so we are able to directly simulate S_T to perform Monte Carlo estimation for European options with a payoff $f(S_T)$.

Euler-Maruyama path simulation

In more general cases, the scalar SDE

$$\mathrm{d}S_t = a(S_t, t)\ \mathrm{d}t + b(S_t, t)\ \mathrm{d}W_t$$

can be approximated using the Euler-Maruyama discretisation

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

Here h is the timestep, $\widehat{S}_n$ is the approximation to S_{nh} and the ΔW_n are i.i.d. $N(0, h)$ Brownian increments.

Euler-Maruyama method

For ODEs, the forward Euler method has $O(h)$ accuracy, and other more accurate methods are usually preferred.

However, SDEs are very much harder to approximate so the Euler-Maruyama method is used widely in practice.

Numerical analysis is also very difficult and even the definition of “accuracy” is tricky.

Weak convergence

In finance applications, we are mostly concerned with **weak** errors, the error in the expected payoff due to using a finite timestep h.

For a European payoff $f(S_T)$, the weak error is

$$\mathbb{E}[f(S_T)] - \mathbb{E}[f(\widehat{S}_M)]$$

where $M = T/h$, and for a path-dependent option it is

$$\mathbb{E}[f(S)] - \mathbb{E}[\widehat{f}(\widehat{S})]$$

where $f(S)$ is a function of the entire path S_t, and $\widehat{f}(\widehat{S})$ is a corresponding approximation using the whole discrete path.

Weak convergence

Key theoretical result (Bally and Talay, 1995):

If $p(S)$ is the p.d.f. for S_T and $\widehat{p}(S)$ is the p.d.f. for $\widehat{S}_{T/h}$ computed using the Euler-Maruyama approximation, then under certain conditions on $a(S,t)$ and $b(S,t)$

$$p(S) - \widehat{p}(S) = O(h)$$

and hence

$$\mathbb{E}[f(S_T)] - \mathbb{E}[f(\widehat{S}_{T/h})] = O(h)$$

This holds even for digital options with discontinuous payoffs $f(S)$ – earlier theory covered only European options such as put and call options with Lipschitz payoffs.

Weak convergence

Numerical demonstration: Geometric Brownian Motion

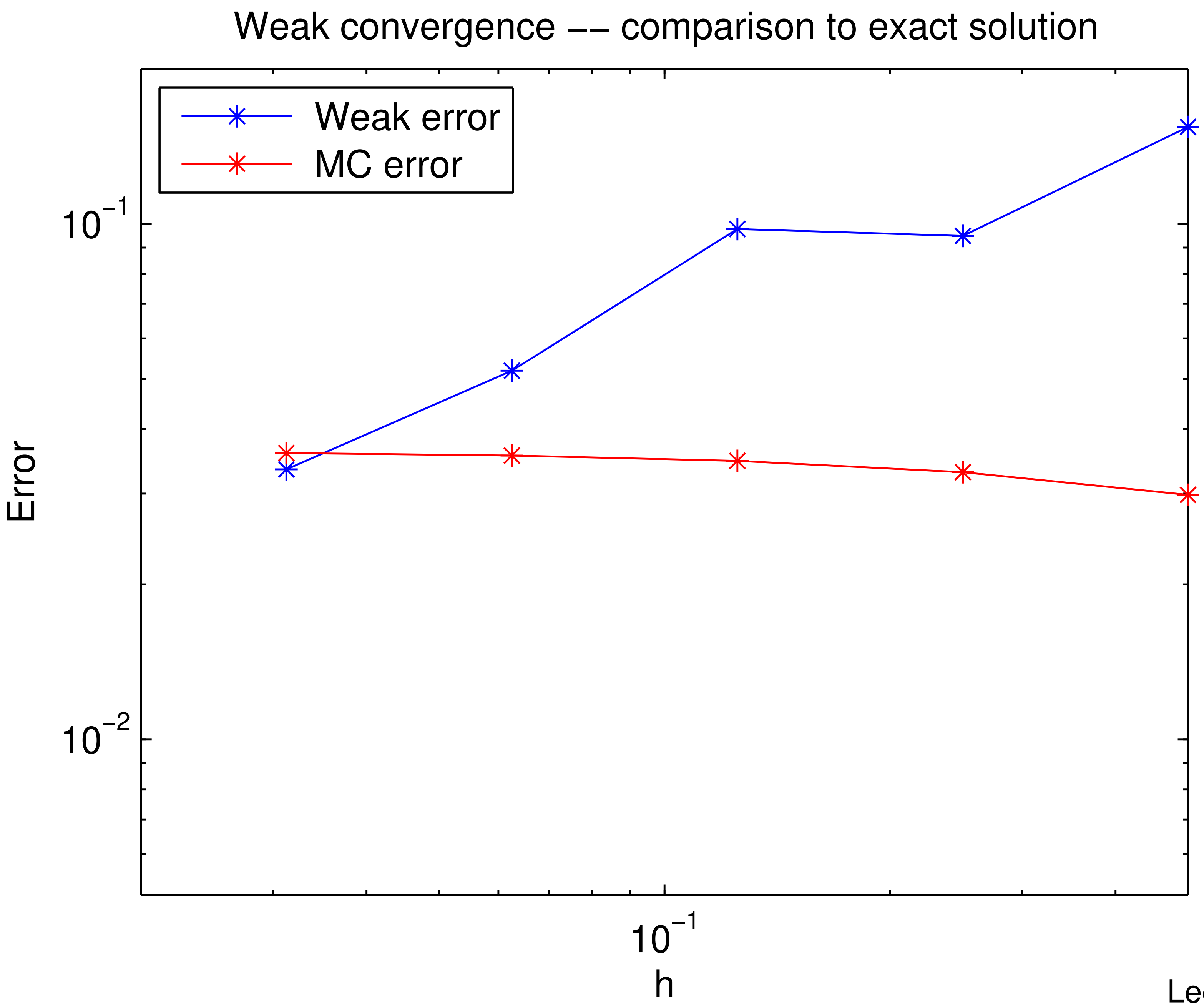
$$\mathrm{d}S = r\, S\, \mathrm{d}t + \sigma\, S\, \mathrm{d}W$$

$r = 0.05,\ \sigma = 0.5,\ T = 1$

European call: $S_0 = 100, K = 110$.

Plot shows weak error versus analytic expectation when using 10^8 paths, and also Monte Carlo error (3 standard deviations)

Weak convergence



Weak convergence

Previous plot showed difference between exact expectation and numerical approximation.

What if the exact solution is unknown? Compare approximations with timesteps h and $2h$.

If

$$\mathbb{E}[f(S_T)] - \mathbb{E}[f(\widehat{S}^h_{T/h})] \approx a\, h$$

then

$$\mathbb{E}[f(S_T)] - \mathbb{E}[f(\widehat{S}^{2h}_{T/2h})] \approx 2\, a\, h$$

and so

$$\mathbb{E}[f(\widehat{S}^h_{T/h})] - \mathbb{E}[f(\widehat{S}^{2h}_{T/2h})] \approx a\, h$$

Weak convergence

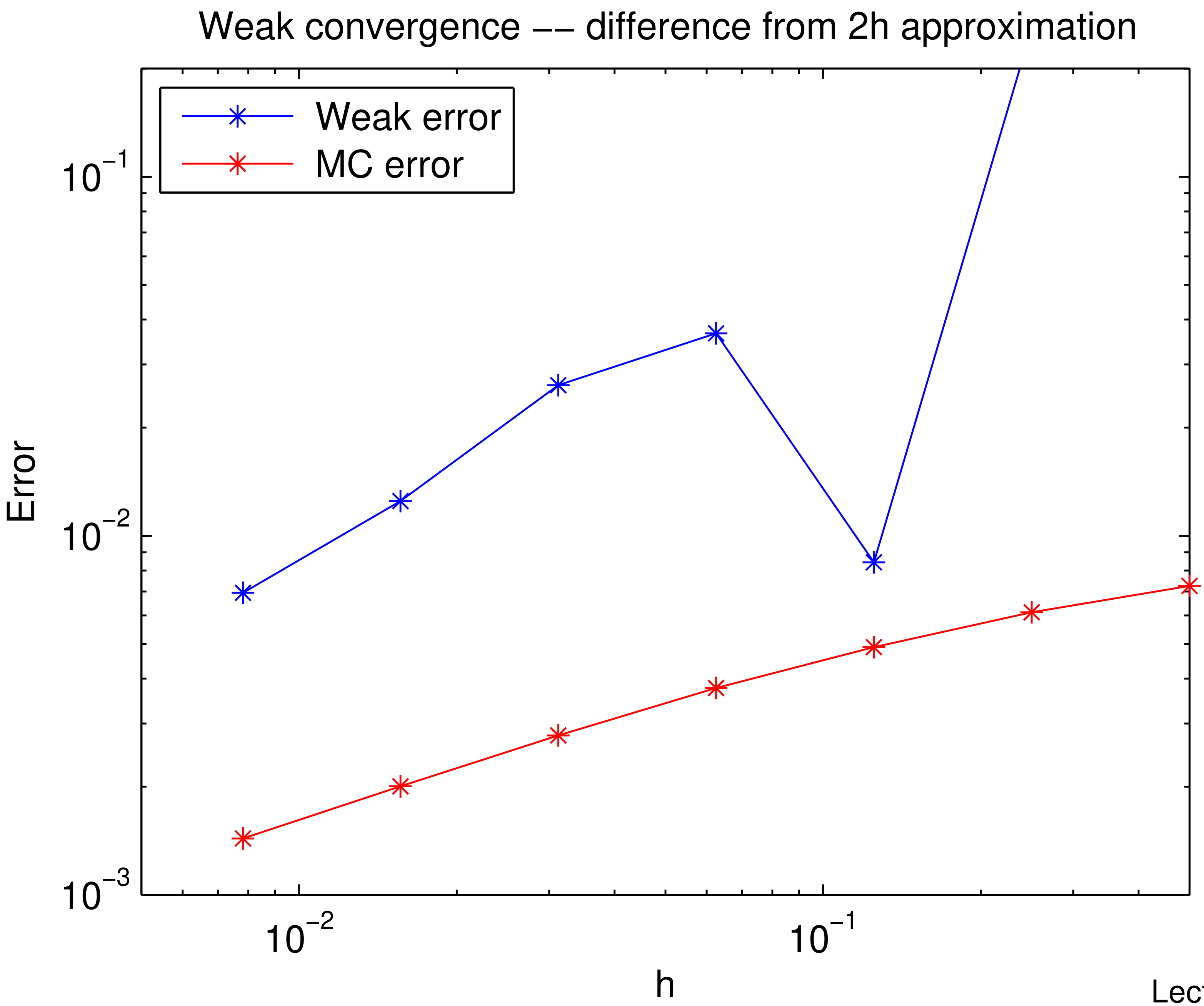
To minimise the number of paths that need to be simulated, we use **same** driving Brownian path when doing $2h$ and h approximations.

i.e. take Brownian increments for h simulation and sum in pairs to get Brownian increments for $2h$ simulation.

The variance is lower because the h and $2h$ paths are close to each other (**strong** convergence).

(We won't cover this, but this forms the basis for the Multilevel Monte Carlo method (Giles, 2006))

Weak convergence



Mean Square Error

Question: how do we choose

- the number of timesteps (to reduce the weak error)
- the number of paths (to reduce the Monte Carlo sampling error)

If the true option value is $$V = \mathbb{E}[f]$$

and the discrete approximation is $$\widehat{V} = \mathbb{E}[\widehat{f}]$$

and the Monte Carlo estimate is $$\widehat{Y} = \frac{1}{N} \sum_{i=1}^{N} \widehat{f}^{(i)}$$

then . . .

Mean Square Error

...the Mean Square Error is

$$
\begin{aligned}
\mathbb{E}\left[\left(\widehat{Y}-V\right)^2\right] &= \mathbb{V}\left[\widehat{Y}-V\right] + \left(\mathbb{E}[\widehat{Y}-V]\right)^2 \\
&= \mathbb{V}[\widehat{Y}] + \left(\mathbb{E}[\widehat{Y}]-V\right)^2 \\
&= N^{-1}\mathbb{V}[\widehat{f}] + \left(\mathbb{E}[\widehat{f}]-\mathbb{E}[f]\right)^2
\end{aligned}
$$

- first term is due to the variance of estimator
- second term is square of bias due to weak error

Mean Square Error

If there are M timesteps, the computational cost is proportional to $C = MN$ and the MSE is approximately

$$a\, N^{-1} + b\, M^{-2} = a\, N^{-1} + b\, C^{-2} N^2.$$

For a fixed computational cost, this is a minimum when

$$N = \left(\frac{a\, C^2}{2\, b} \right)^{1/3}, \quad M = \left(\frac{2\, b\, C}{a} \right)^{1/3},$$

and hence

$$a\, N^{-1} = \left(\frac{2\, a^2 b}{C^2} \right)^{1/3}, \quad b\, M^{-2} = \left(\frac{a^2 b}{4\, C^2} \right)^{1/3},$$

so the MC term is twice as big as the bias term.

Path-dependent Options

For European options, Euler-Maruyama method has $O(h)$ weak convergence.

However, for some path-dependent options it may give only $O(\sqrt{h})$ weak convergence, unless the numerical payoff is constructed carefully.

Barrier option

A down-and-out call option has discounted payoff

$$\exp(-rT)\ (S_T - K)^+ \mathbf{1}_{\min_t S(t) > B}$$

i.e. it is like a standard call option except that it pays nothing if the minimum value drops below the barrier B.

The natural numerical discretisation of this is

$$f = \exp(-rT)\ (\widehat{S}_M - K)^+ \mathbf{1}_{\min_n \widehat{S}_n > B}$$

Barrier option

Numerical demonstration: Geometric Brownian Motion

$$\mathrm{d}S_t = r\,S_t\,\mathrm{d}t + \sigma\,S_t\,\mathrm{d}W_t$$

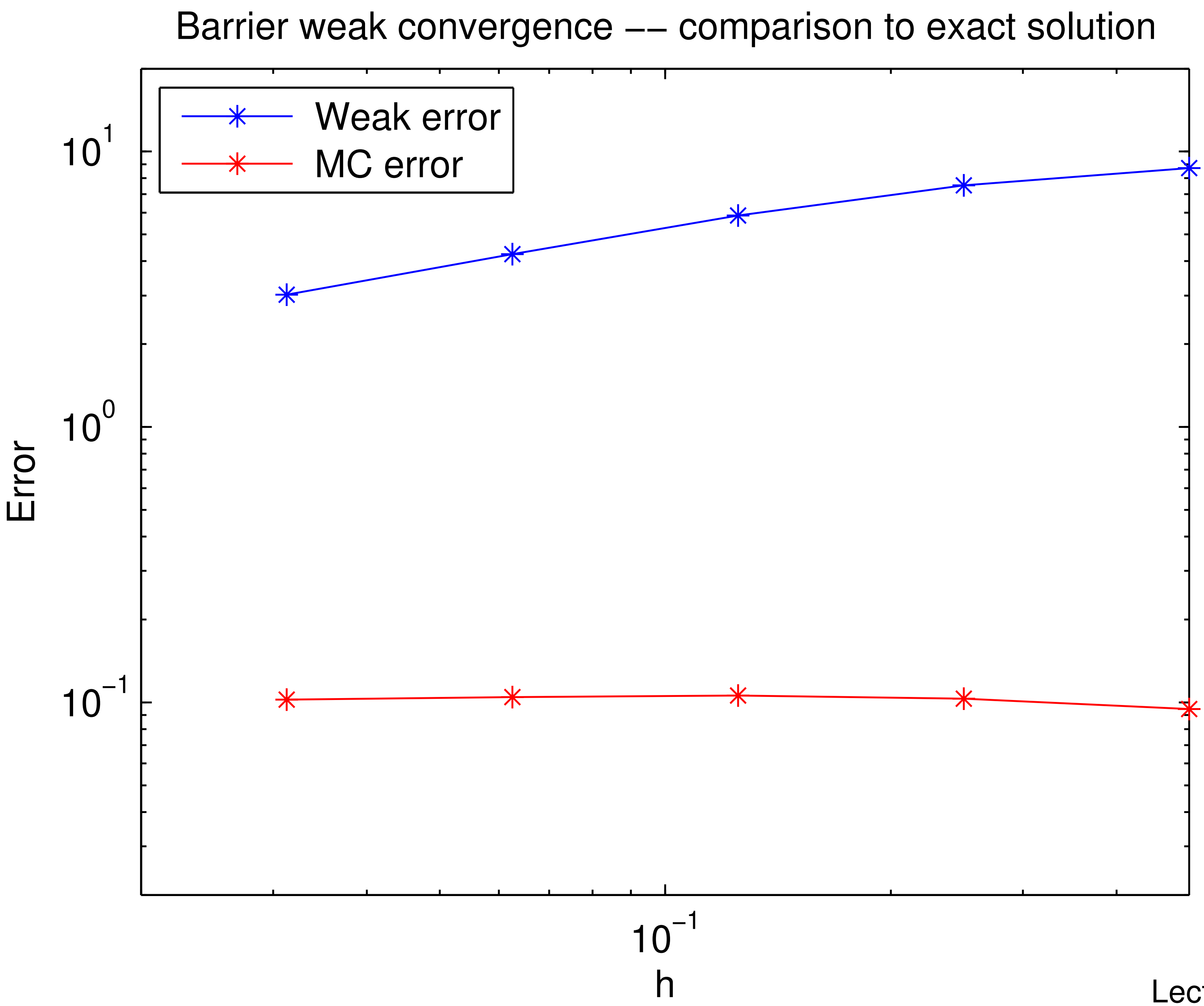
$r = 0.05,\ \sigma = 0.5,\ T = 1$

Down-and-out call: $S_0 = 100, K = 110, B = 90$.

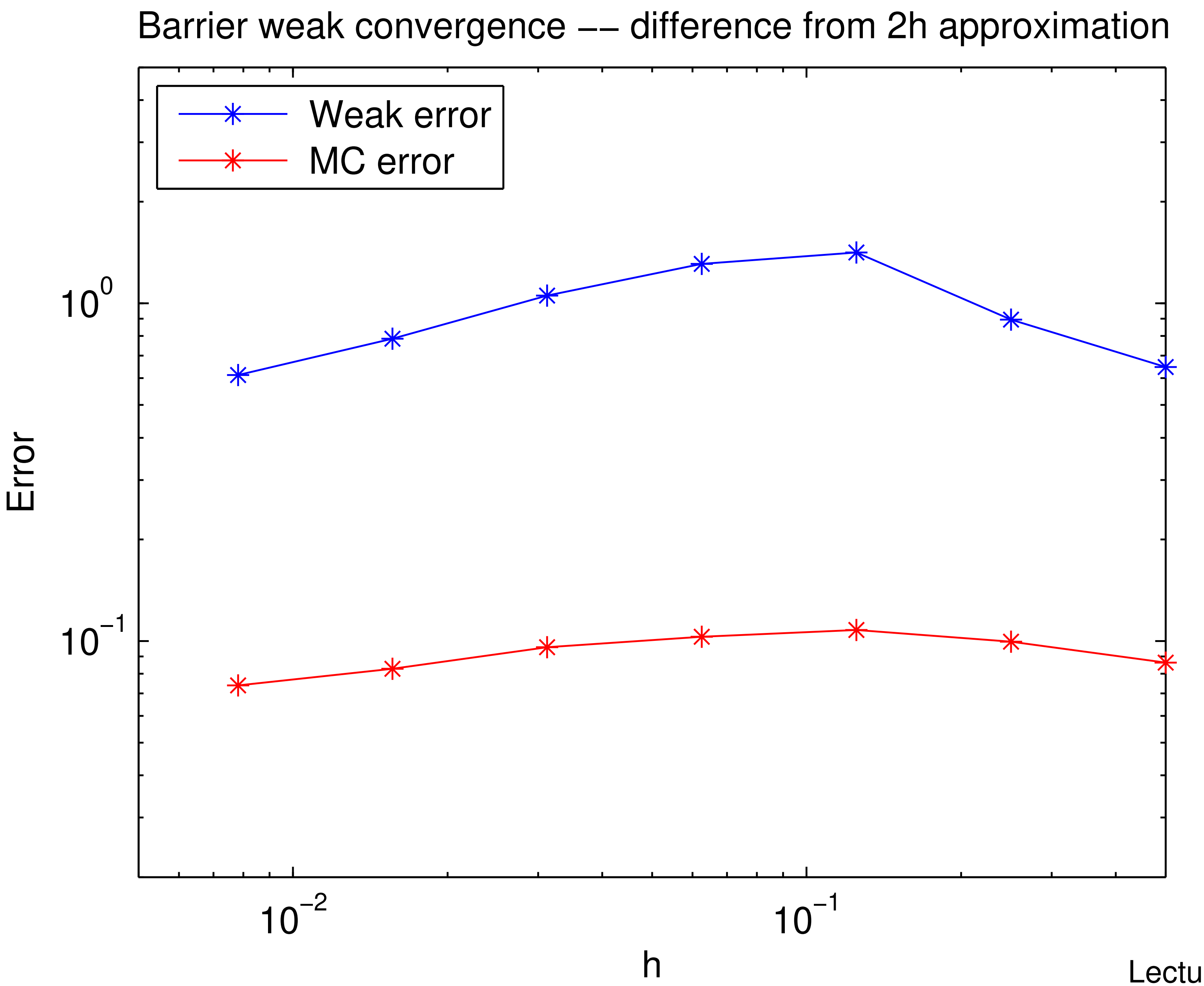
Plots shows weak error versus analytic expectation using 10^6 paths, and difference from $2h$ approximation using 10^5 paths.

(We don't need as many paths as before because the weak errors are much larger in this case.)

Barrier option



Barrier option



Lookback option

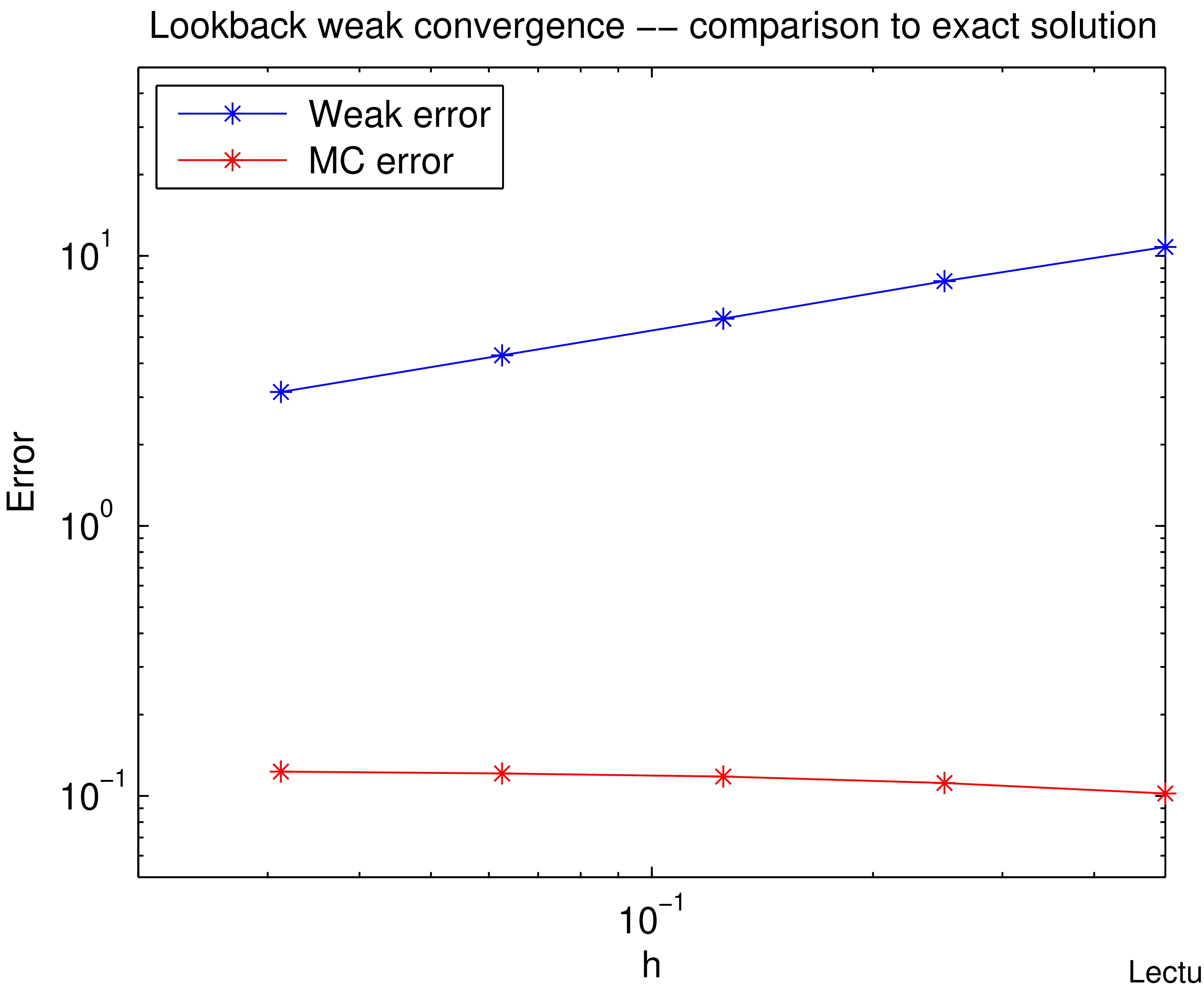
A floating-strike lookback call option has discounted payoff

$$\exp(-rT)\ \left(S_T - \min_{[0,T]} S_t \right)$$

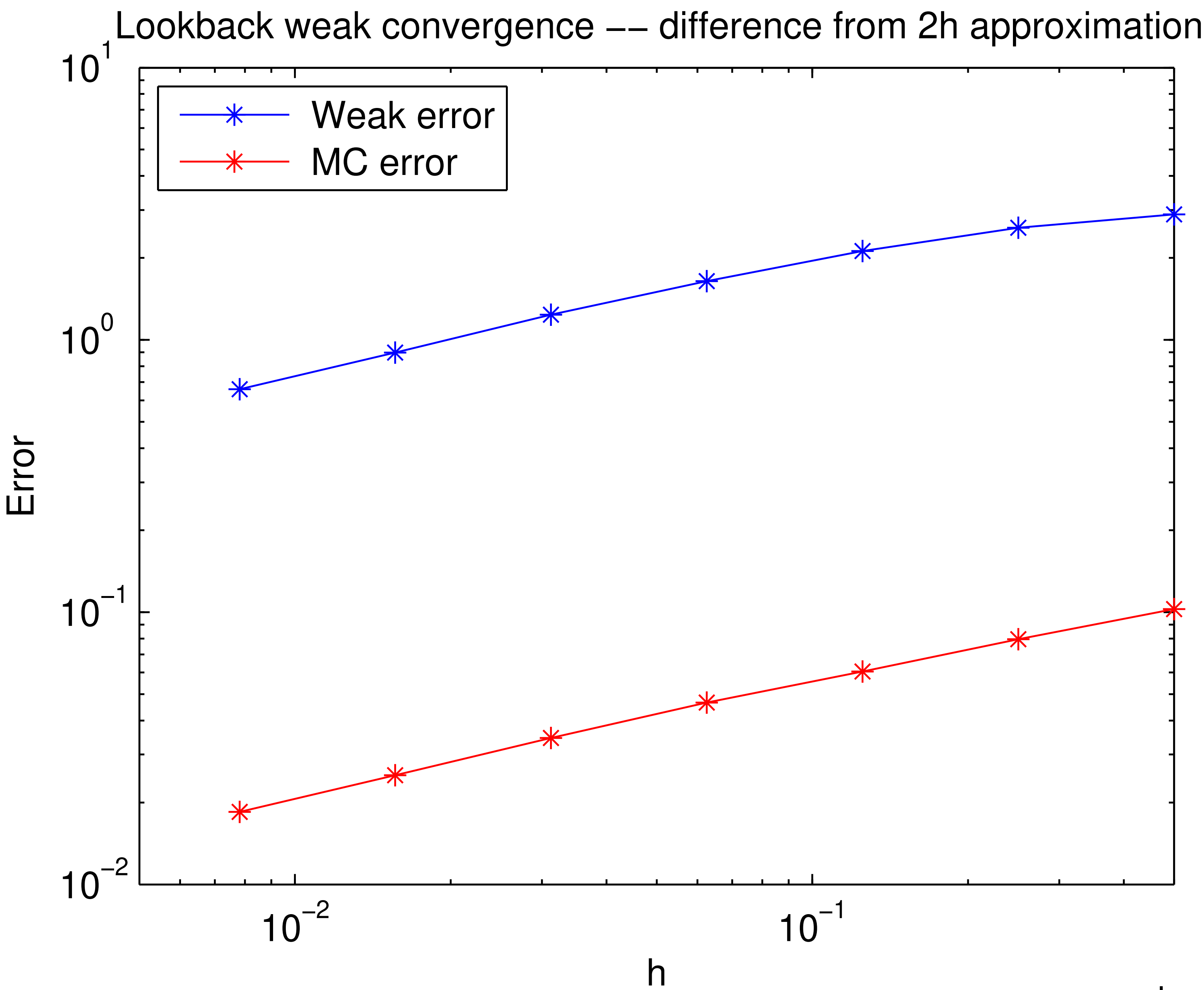
The natural numerical discretisation of this is

$$f = \exp(-rT)\ \left(\widehat{S}_M - \min_n \widehat{S}_n \right)$$

Lookback option



Lookback option



Brownian Bridge

To recover $O(h)$ weak convergence we first need some theory.

Consider simple Brownian motion

$$\mathrm{d}S_t = a\,\mathrm{d}t + b\,\mathrm{d}W_t$$

with constant $a,\ b$ and initial data $S_0\!=\!0$.

Question: given S_T, what is conditional probability density for $S_{T/2}$?

Conditional probability

With discrete probabilities,

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Similarly, with probability density functions

$$p_1(x|y) = \frac{p_2(x,y)}{p_3(y)}$$

where

- $p_1(x|y)$ is the conditional p.d.f. for x, given y
- $p_2(x,y)$ is the joint probability density function for x, y
- $p_3(y)$ is the probability density function for y

Brownian bridge

In our case,

$$y \equiv S_T, \quad x \equiv S_{T/2}$$

$$\begin{aligned}
p_2(x,y) &= \frac{1}{\sqrt{\pi\,T}\;b}\;\exp\left(-\,\frac{(x-aT/2)^2}{b^2\,T}\right)\\
&\times\;\frac{1}{\sqrt{\pi\,T}\;b}\;\exp\left(-\,\frac{(y-x-aT/2)^2}{b^2\,T}\right)\\
p_3(y) &= \frac{1}{\sqrt{2\pi\,T}\;b}\;\exp\left(-\,\frac{(y-aT)^2}{2\,b^2\,T}\right)\\
\Longrightarrow p_1(x|y) &= \frac{1}{\sqrt{\pi\,T/2}\;b}\;\exp\left(-\,\frac{(x-y/2)^2}{b^2\,T/2}\right)
\end{aligned}$$

Hence, x is Normally distributed with mean $y/2$ and variance $b^2T/4$.

Brownian bridge

Extending this to a particular timestep with endpoints S_n and S_{n+1}, conditional on these the mid-point is Normally distributed with mean

$$\frac{1}{2}\left(S_n + S_{n+1}\right)$$

and variance $b^2 h/4$.

We can take a sample from this conditional p.d.f. and then repeat the process, recursively bisecting each interval to fill in more and more detail.

Note: the drift a is irrelevant, given the two endpoints. Because of this, we will take $a=0$ in the next bit of theory.

Barrier crossing

Consider zero drift Brownian motion with $S_0 > 0$.

If the path S_t hits a barrier at 0, it is equally likely thereafter to go up or down. Hence, by symmetry, for $s > 0$, the p.d.f. for paths with $S_T = s$ after hitting the barrier is equal to the p.d.f. for paths with $S_T = -s$.

Thus, for $S_T > 0$,

$$
\begin{aligned}
P(\text{hit barrier} \,|\, S_T) &= \frac{\exp\left(-\frac{(-S_T - S_0)^2}{2b^2 T}\right)}{\exp\left(-\frac{(S_T - S_0)^2}{2b^2 T}\right)} \\
&= \exp\left(-\frac{2\, S_T\, S_0}{b^2 T}\right)
\end{aligned}
$$

Barrier crossing

For a timestep $[t_n, t_{n+1}]$ and non-zero barrier B this generalises to

$$P(\text{hit barrier} \,|\, S_n, S_{n+1} > B) = \exp\left(-\frac{2\,(S_{n+1}\!-\!B)\,(S_n\!-\!B)}{b^2 h} \right)$$

This can also be viewed as the cumulative probability $P(S_{min} < B)$ where $S_{min} = \min\limits_{[t_n,t_{n+1}]} S(t)$.

Since this is uniformly distributed on $[0,1]$ we can equate this to a uniform $[0,1]$ random variable U_n and solve to get

$$S_{min} = \tfrac{1}{2}\left(S_{n+1} + S_n - \sqrt{(S_{n+1}\!-\!S_n)^2 - 2\, b^2 h \, \log U_n} \right)$$

Barrier crossing

For a barrier above, we have

$$P(\text{hit barrier} \,|\, S_n, S_{n+1} < B) = \exp\left(- \frac{2\,(B - S_{n+1})\,(B - S_n)}{b^2 h} \right)$$

and hence

$$S_{max} = \tfrac{1}{2}\left(S_{n+1} + S_n + \sqrt{(S_{n+1} - S_n)^2 - 2\, b^2 h\, \log U_n} \right)$$

where U_n is again a uniform $[0,1]$ random variable.

Barrier option

Returning now to the barrier option, how do we define the numerical payoff $\widehat{f}(\widehat{S})$?

First, calculate $\widehat{S}_n$ as usual using Euler-Maruyama method.

Second, two alternatives:

- use (approximate) probability of crossing the barrier
- directly sample (approximately) the minimum in each timestep

Barrier option

Alternative 1: treating the drift and volatility as being approximately constant within each timestep, the probability of having crossed the barrier within timestep n is

$$P_n = \exp\left(- \frac{2\,(\widehat{S}_{n+1}\!-\!B)^+\,(\widehat{S}_n\!-\!B)^+}{b^2(\widehat{S}_n, t_n)\ h} \right)$$

Probability at end of <u>not</u> having crossed barrier is $\prod_n (1 - P_n)$ and so the payoff is

$$\widehat{f}(\widehat{S}) = \exp(-rT)\ (\widehat{S}_M - K)^+\ \prod_n (1 - P_n).$$

I prefer this approach because it is differentiable – good for Greeks

Barrier option

Alternative 2: again treating the drift and volatility as being approximately constant within each timestep, define the minimum within timestep n as

$$\widehat{M}_n = \frac{1}{2}\left(\widehat{S}_{n+1} + \widehat{S}_n - \sqrt{(\widehat{S}_{n+1} - \widehat{S}_n)^2 - 2\, b^2(\widehat{S}_n, t_n)\, h\, \log U_n}\right)$$

where the U_n are i.i.d. uniform $[0,1]$ random variables.

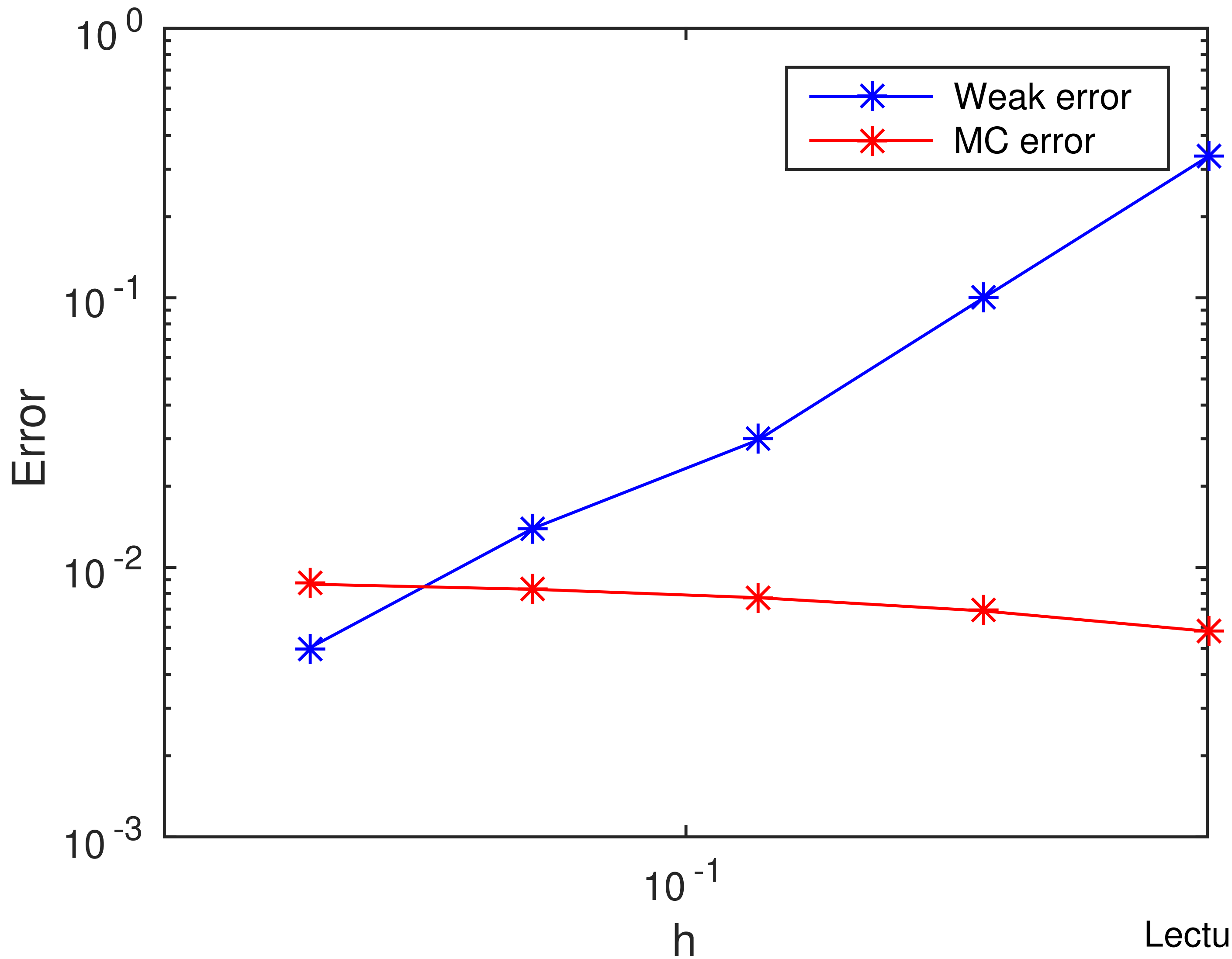
The payoff is then

$$\widehat{f}(\widehat{S}) = \exp(-rT)\; (\widehat{S}_M - K)^+\; \mathbf{1}_{\min_n \widehat{M}_n > B}$$

With this approach one can stop the path calculation as soon as one $\widehat{M}_n$ drops below B.

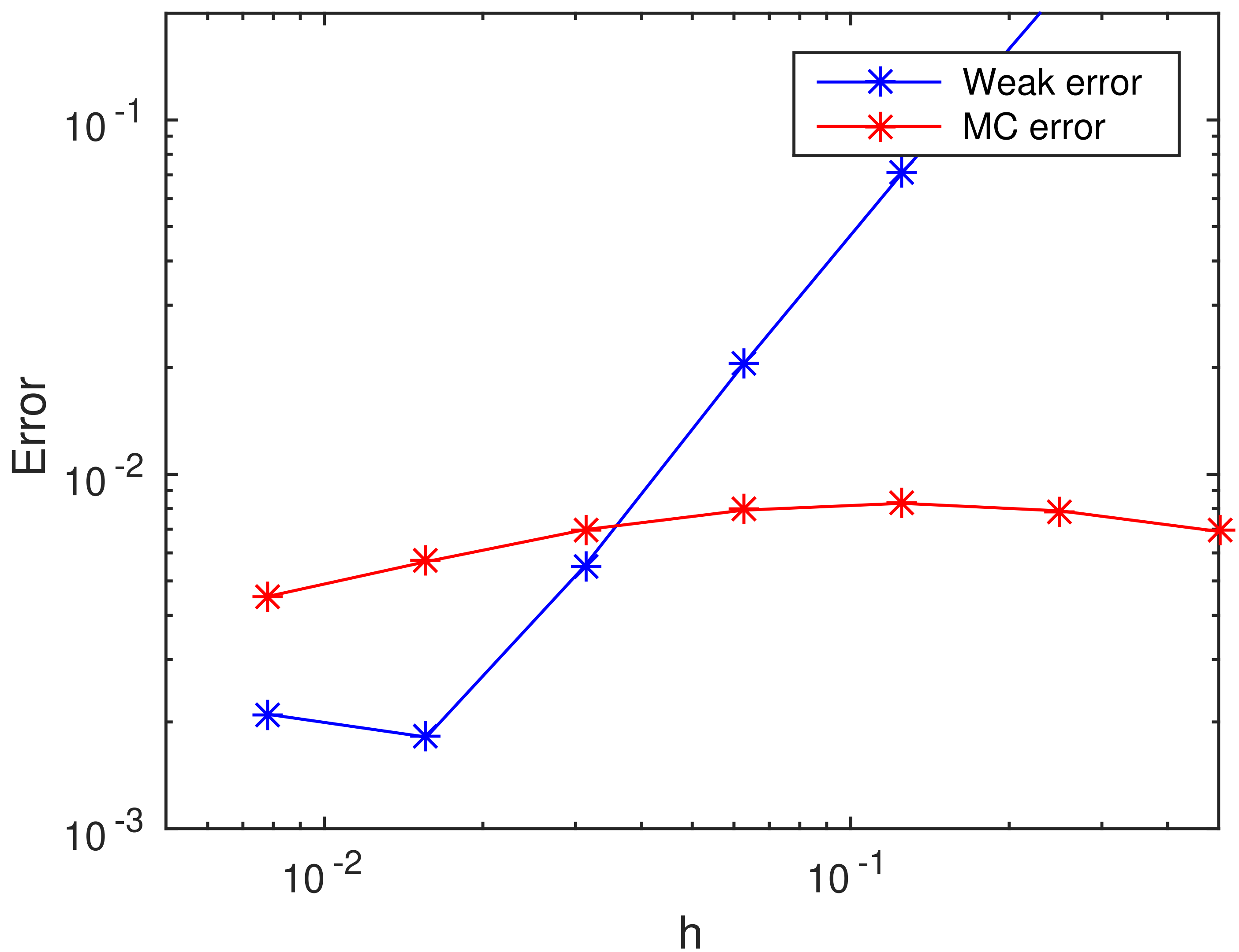
Weak convergence

Barrier: comparison to solution



Weak convergence

Barrier: h versus $2h$ solution



Lookback option

This is treated in a similar way to Alternative 2 for the barrier option.

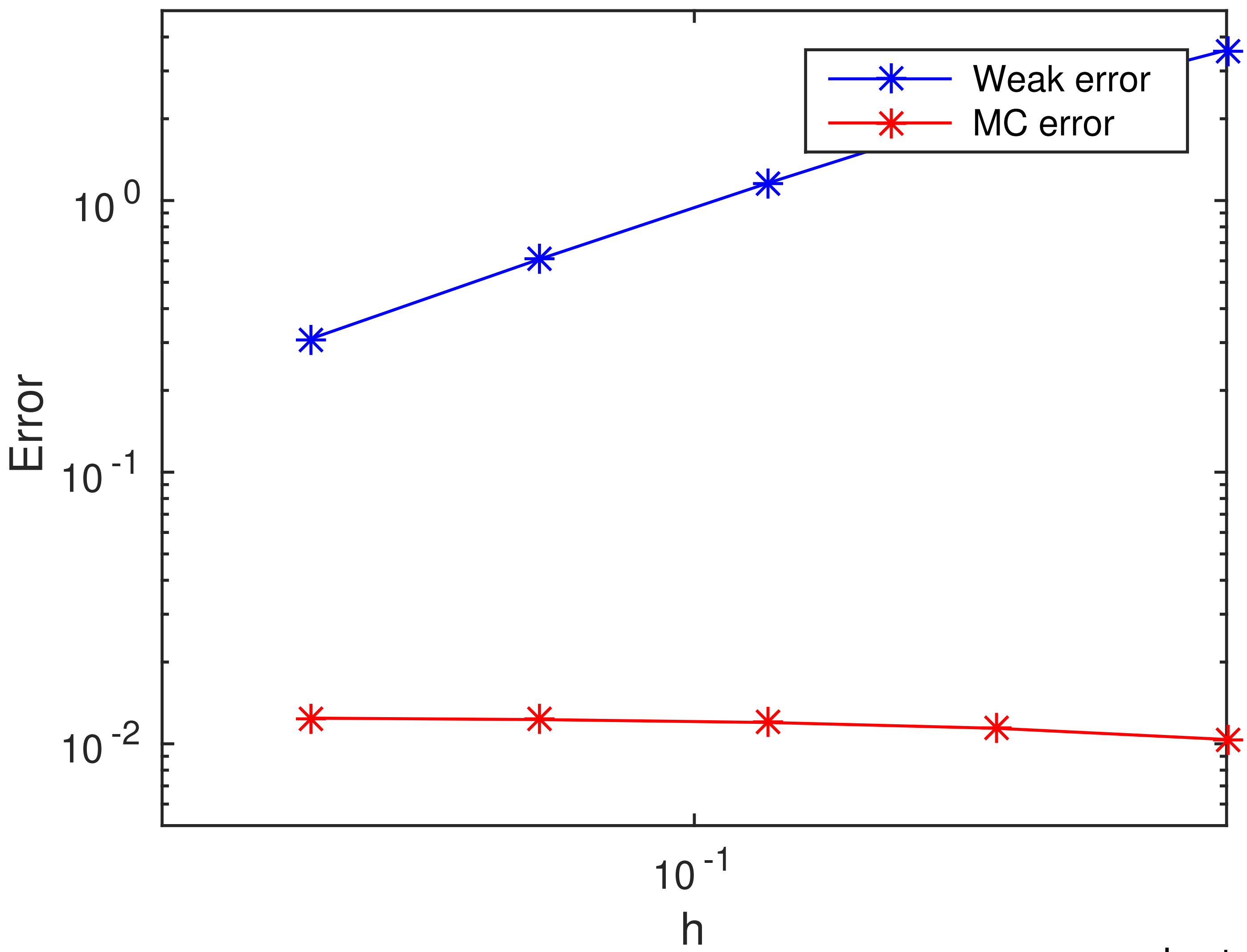
We construct a minimum $\widehat{M}_n$ within each timestep and then the payoff is

$$\widehat{f}(\widehat{S}) = \exp(-rT)\ \left(\widehat{S}_M - \min_n \widehat{M}_n\right)$$

This is differentiable, so good for Greeks – unlike Alternative 2 for the barrier option.

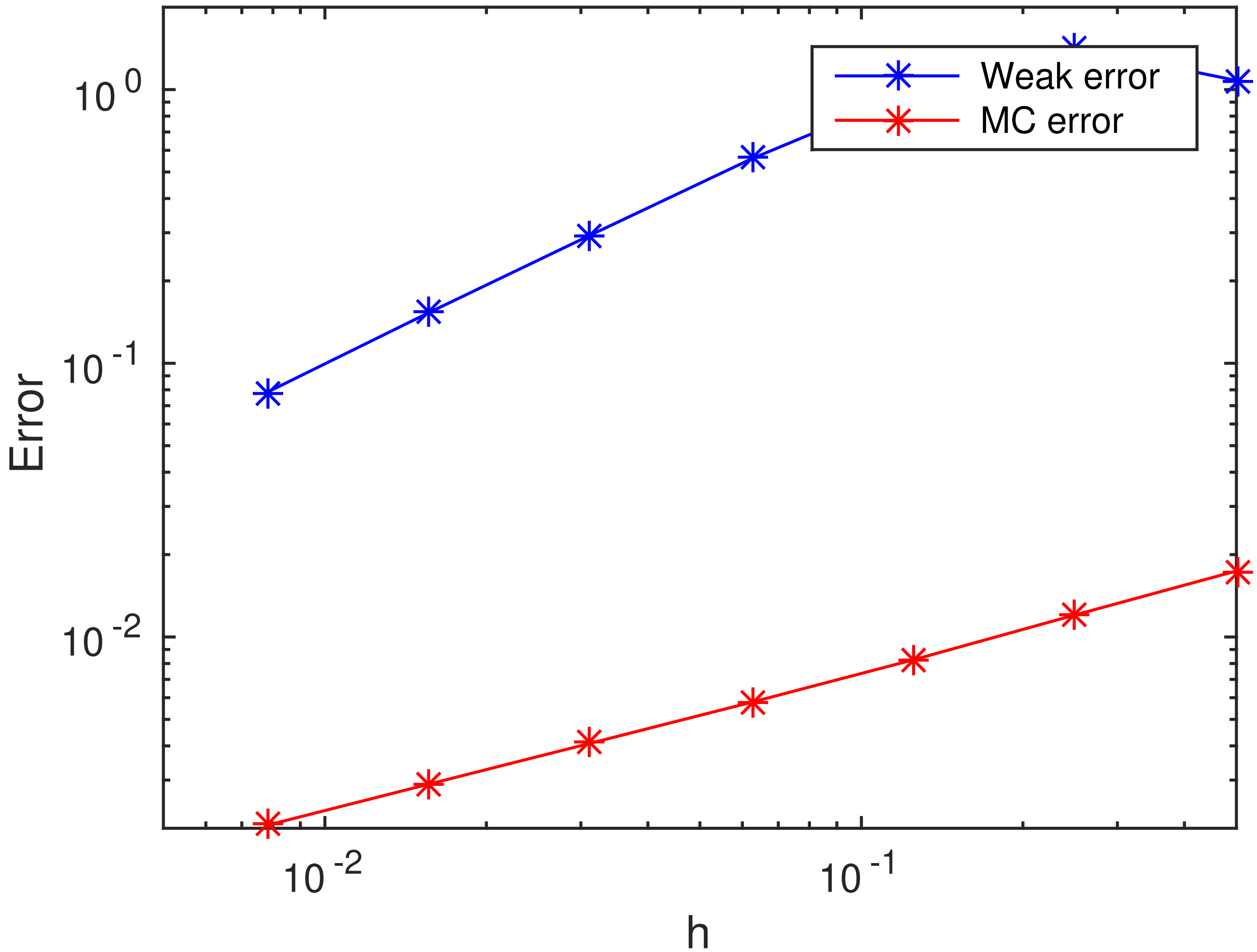
Weak convergence

Lookback: comparison to true solution



Weak convergence

Lookback: h versus $2h$ solution



Final Words

- Euler-Maruyama gives $O(h)$ weak convergence for European options
- Mean Square Error analysis shows how to balance weak errors and Monte Carlo sampling errors
- “natural” approximation of barrier and lookback options leads to poor $O(\sqrt{h})$ weak convergence due to $O(\sqrt{h})$ path variation within each timestep
- improved treatment based on Brownian bridge theory approximates behaviour within timestep as simple Brownian motion with constant drift and volatility
  – gives $O(h)$ weak convergence

Quasi-Monte Carlo

You have previously learned about Quasi-Monte Carlo for European options based on Geometric Brownian Motion, so the underlying can be directly simulated at the final time T.

Now consider path-dependent options which require us to simulate the underlying asset(s) by approximating the SDE.

Same ingredients:

- Sobol or lattice rule quasi-uniform generators
- $Z = \Phi^{-1}(U)$ to convert quasi-uniform random numbers to quasi-Normal random numbers
- PCA (Principal Component Analysis) to best use QMC inputs for multi-dimensional applications
- randomised QMC to regain confidence interval

Quasi-Monte Carlo

For a scalar SDE, using Euler-Maruyama approximation

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

with $\Delta W_n = \sqrt{h}\, Z_n$, can express expectation as a multi-dimensional integral with respect to unit Normal inputs

$$V = \mathbb{E}[\widehat{f}(\widehat{S})] = \int \widehat{f}(\widehat{S})\ \phi(Z)\ \mathrm{d}Z$$

where $\phi(Z)$ is multi-dimensional unit Normal p.d.f.

Putting $Z_n = \Phi^{-1}(U_n)$ turns this into an integral over a M-dimensional hypercube

$$V = \mathbb{E}[\widehat{f}(\widehat{S})] = \int \widehat{f}(\widehat{S})\ \mathrm{d}U$$

Quasi-Monte Carlo

This is then approximated as

$$N^{-1} \sum_n \widehat{f}(\widehat{S}^{(n)})$$

and each path calculation involves the computations

$$U \;\rightarrow\; Z \;\rightarrow\; \Delta W \;\rightarrow\; \widehat{S} \;\rightarrow\; \widehat{f}$$

The key step here is the second, how best to convert the vector Z into the vector ΔW.

With standard Monte Carlo, as long as ΔW has the correct distribution, how it is generated is irrelevant, but with QMC it does matter.

Quasi-Monte Carlo

For a scalar Brownian motion $W(t)$ with $W(0)\!=\!0$, defining $W_n\!=\!W(nh)$, each W_n is Normally distributed and for $j \geq k$

$$\mathbb{E}[W_j\, W_k] = \mathbb{E}[W_k^2] + \mathbb{E}[(W_j\!-\!W_k)\, W_k] = t_k$$

since $W_j\!-\!W_k$ is independent of W_k. Hence, the covariance matrix for W is Ω with elements

$$\Omega_{j,k} = \min(t_j, t_k)$$

Given a vector of uncorrelated units Normals Z, can define W as

$$W = L\, Z,$$

where

$$\Omega = \mathbb{E}[W W^T] = \mathbb{E}[L\, Z\, Z^T L^T] = L\, L^T.$$

Quasi-Monte Carlo

The task now is to find a matrix L such that

$$L\,L^T = \Omega = h \begin{pmatrix} 1 & 1 & \ldots & 1 & 1 \\ 1 & 2 & \ldots & 2 & 2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 2 & \ldots & M\!-\!1 & M\!-\!1 \\ 1 & 2 & \ldots & M\!-\!1 & M \end{pmatrix}$$

We will consider 3 possibilities:

- Cholesky factorisation
- PCA
- Brownian Bridge treatment

Cholesky factorisation

The Cholesky factorisation gives

$$L = \sqrt{h} \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 \\ 1 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 1 & \ldots & 1 & 0 \\ 1 & 1 & \ldots & 1 & 1 \end{pmatrix}$$

and hence

$$W_n = \sum_{m=1}^{n} \sqrt{h}\, Z_m \quad \Longrightarrow \quad \Delta W_n = W_n - W_{n-1} = \sqrt{h}\, Z_n$$

i.e. standard MC approach

PCA construction

The PCA construction uses

$$L = U\,\Lambda^{1/2} = \left(\; U_1 \;\middle|\; U_2 \;\middle|\; \ldots \; \right) \begin{pmatrix} \lambda_1^{1/2} & & \\ & \lambda_2^{1/2} & \\ & & \ldots \end{pmatrix}$$

with the eigenvalues λ_n and eigenvectors U_n arranged in descending order, from largest to smallest.

Numerical computation of the eigenvalues and eigenvectors is costly for large numbers of timesteps, so instead use theory due to Åkesson and Lehoczky (1998)

PCA construction

It is easily verified that

$$\Omega^{-1} = h^{-1} \begin{pmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ldots & \ldots & \ldots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 1 \end{pmatrix}.$$

This looks like the finite difference operator approximating a second derivative, and so the eigenvectors are Fourier modes.

PCA construction

The eigenvectors of both Ω^{-1} and Ω are

$$(U_m)_n = \frac{2}{\sqrt{2M+1}} \sin\left(\frac{(2m-1)\,n\pi}{2M+1}\right)$$

and the eigenvalues of Ω are the reciprocal of those of Ω^{-1},

$$\lambda_m = \frac{h}{4}\left(\sin\left(\frac{(2m-1)\,\pi}{2\,(2M+1)}\right)\right)^{-2}$$

Because the eigenvectors are Fourier modes, an efficient FFT transform can be used (Scheicher, 2006) to compute

$$L\,Z \;=\; U\,\left(\Lambda^{1/2}\,Z\right) \;= \sum_m (\sqrt{\lambda_m}\,Z_m)\,U_m$$

Brownian Bridge construction

The Brownian Bridge construction uses the theory already developed.

The final Brownian value is constructed using Z_1:

$$W_M = \sqrt{T}\, Z_1$$

Conditional on this, the midpoint value $W_{M/2}$ is Normally distributed with mean $\frac{1}{2}W_M$ and variance $T/4$, and so can be constructed as

$$W_{M/2} = \tfrac{1}{2}W_M + \sqrt{T/4}\, Z_2$$

Brownian Bridge construction

The quarter and three-quarters points can then be constructed as

$$
\begin{aligned}
W_{M/4} &= \tfrac{1}{2} W_{M/2} + \sqrt{T/8}\, Z_3 \\
W_{3M/4} &= \tfrac{1}{2} (W_{M/2} + W_M) + \sqrt{T/8}\, Z_4
\end{aligned}
$$

and the procedure continued recursively until all Brownian values are defined.

(This assumes M is a power of 2 – if not, the implementation is slightly more complex)

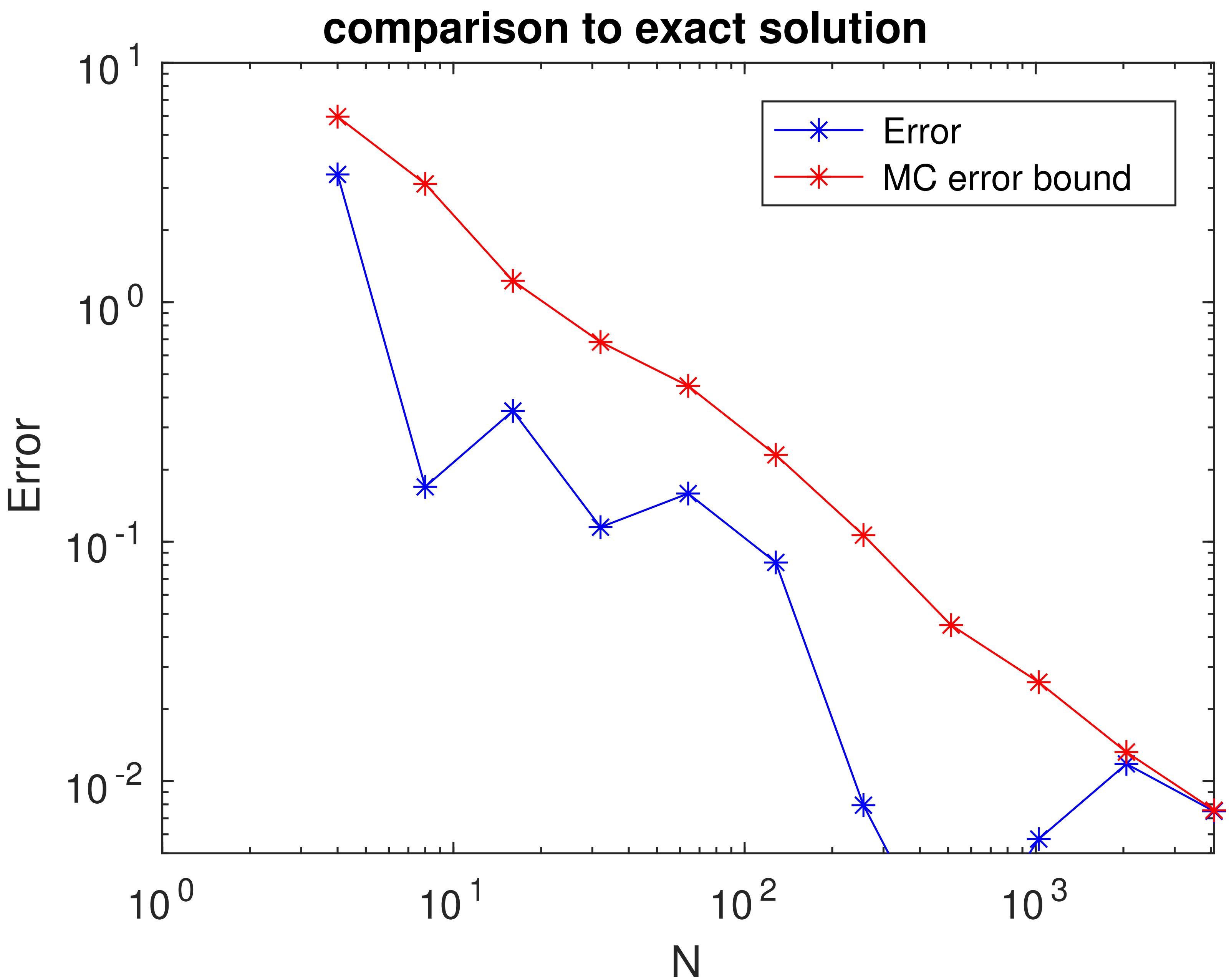
I have a slight preference for this method because it is particularly effective for European options for which S_T is very strongly dependent on $W(T)$.

Numerical results

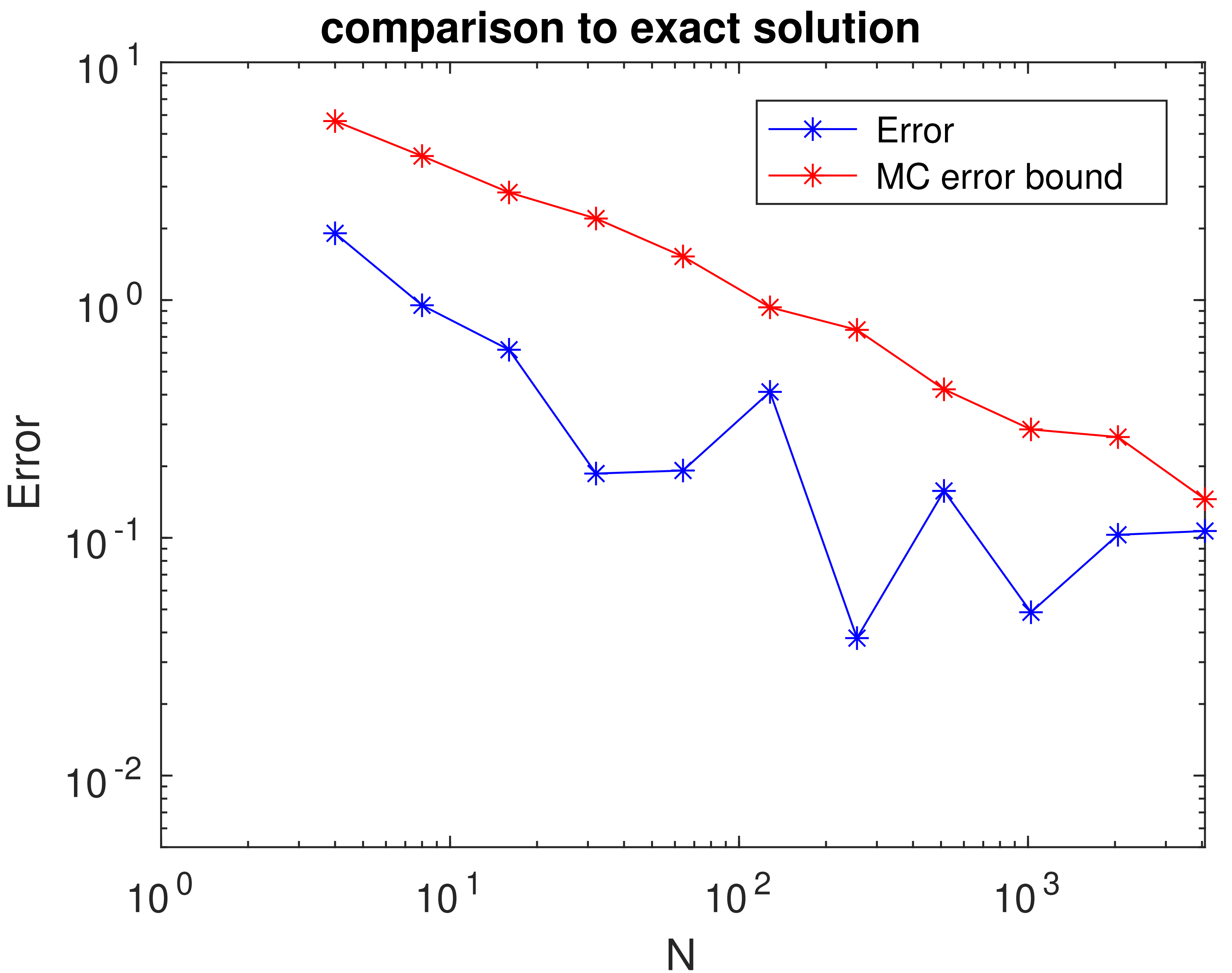
Usual European call test case based on geometric Brownian motion:

- 128 timesteps so weak error is negligible
- comparison between
  - QMC using Brownian Bridge
  - QMC without Brownian Bridge
  - standard MC
- QMC calculations use Sobol points
- all calculations use 64 “sets” of points – for QMC calcs, each has a different random offset
- plots show error and 3 s.d. error bound

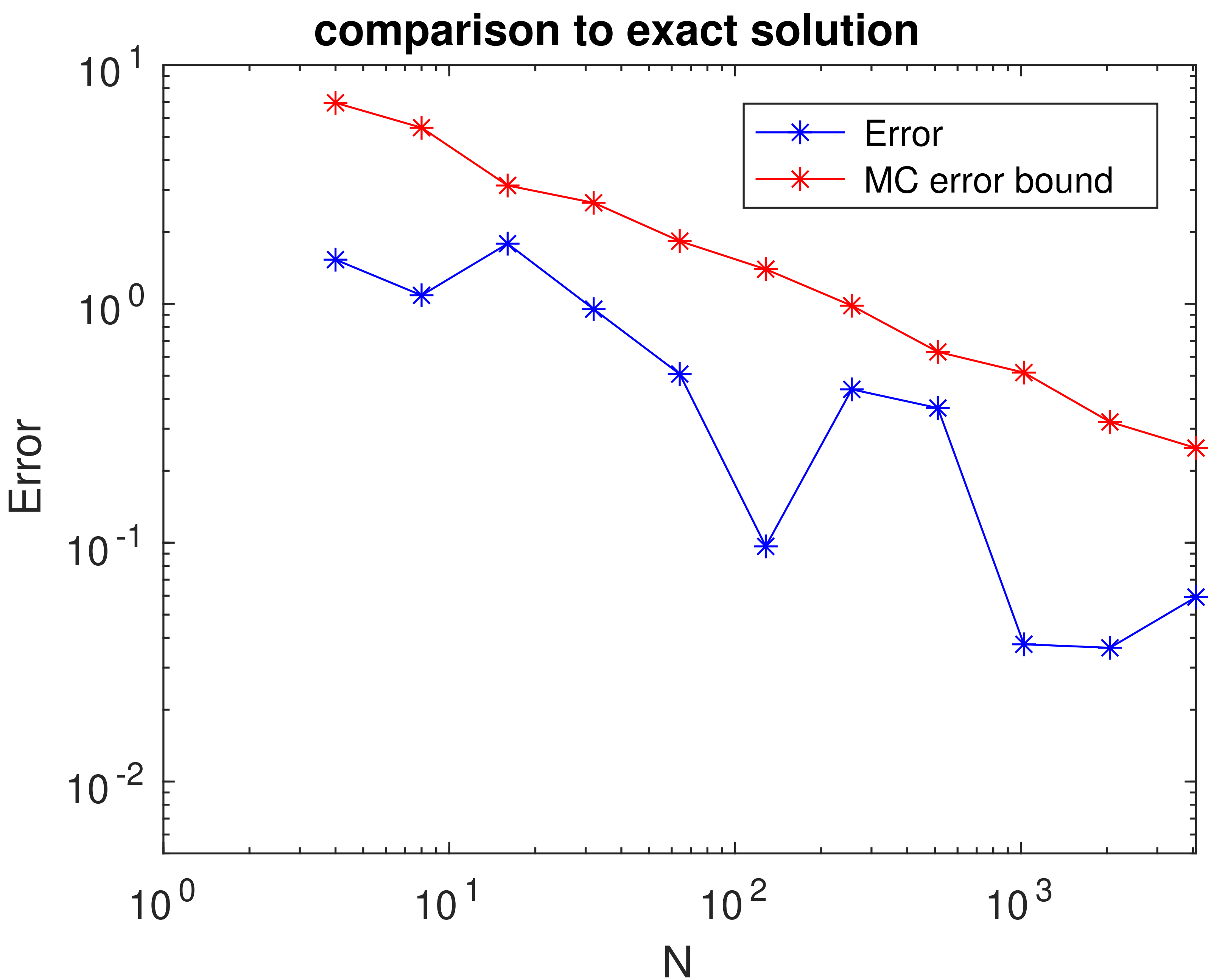
QMC with Brownian Bridge



QMC without Brownian Bridge



Standard Monte Carlo



Final words

- QMC offers large computational savings over the standard Monte Carlo approach
- again advisable to use randomised QMC to regain confidence intervals, at the cost of slightly poorer accuracy
- very important to use PCA or Brownian Bridge construction to create discrete Brownian increments – much better than “standard” approach which is equivalent to Cholesky factorisation of covariance matrix
- for multi-dimensional SDEs, combine Brownian Bridge construction in time, with PCA for correlation between multiple assets.

Monte Carlo Methods

Prof. Mike Giles

mike.giles@maths.ox.ac.uk

Oxford University Mathematical Institute

Greeks

In Monte Carlo applications we don't just want to know the expected discounted value of some payoff

$$V = \mathbb{E}[f(S(T))]$$

We also want to know a whole range of "Greeks" corresponding to first and second derivatives of V with respect to various parameters:

$$\Delta = \frac{\partial V}{\partial S_0}, \qquad \Gamma = \frac{\partial^2 V}{\partial S_0^2},$$

$$\rho = \frac{\partial V}{\partial r}, \qquad \text{Vega} = \frac{\partial V}{\partial \sigma}.$$

Greeks

The Greeks are needed for hedging and risk analysis.

Whereas prices can be obtained to some extent from market prices, simulation is the only way to determine the Greeks.

You have already learned about estimating Greeks by finite differences, so we now discuss 2 alternatives:

- likelihood ratio method
- pathwise sensitivities

Likelihood ratio method

Defining $p(S)$ to the probability density function for the final state S_T, then

$$V = \mathbb{E}[f(S_T)] = \int f(S)\, p(S)\, \mathrm{d}S,$$

$$\Longrightarrow \quad \frac{\partial V}{\partial \theta} = \int f\, \frac{\partial p}{\partial \theta}\, \mathrm{d}S = \int f\, \frac{\partial (\log p)}{\partial \theta}\, p\, \mathrm{d}S = \mathbb{E}\left[f\, \frac{\partial (\log p)}{\partial \theta} \right]$$

The quantity $\dfrac{\partial (\log p)}{\partial \theta}$ is sometimes called the “score function”.

Likelihood ratio method

Note that when $f=1$, we get

$$\frac{\partial}{\partial\theta}\mathbb{E}[1] = 0$$

and therefore

$$\mathbb{E}\left[\frac{\partial(\log p)}{\partial\theta}\right] = 0$$

This is a handy check to make sure we have derived the score function correctly.

Likelihood ratio method

Example: GBM with arbitrary payoff $f(S_T)$.

For the usual Geometric Brownian motion with constants r, σ, the final log-normal probability distribution is

$$p(S) = \frac{1}{S\sigma\sqrt{2\pi T}} \; \exp\left[-\tfrac{1}{2} \left(\frac{\log(S/S_0) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \right)^2 \right]$$

$$\log p = -\log S - \log\sigma - \tfrac{1}{2}\log(2\pi T) - \tfrac{1}{2} \frac{\left(\log(S/S_0) - (r - \frac{1}{2}\sigma^2)T\right)^2}{\sigma^2 T}$$

$$\Longrightarrow \quad \frac{\partial \log p}{\partial S_0} = \frac{\log(S/S_0) - (r - \frac{1}{2}\sigma^2)T}{S_0\sigma^2 T}$$

Likelihood ratio method

Hence

$$\Delta = \mathbb{E}\left[\frac{\log(S/S_0) - (r-\frac{1}{2}\sigma^2)T}{S_0\,\sigma^2 T}\; f(S_T)\right]$$

In the Monte Carlo simulation,

$$\log(S/S_0) - (r-\tfrac{1}{2}\sigma^2)T = \sigma\, W(T)$$

so the expression can be simplified to

$$\Delta = \mathbb{E}\left[\frac{W(T)}{S_0\,\sigma\,T}\; f(S_T)\right]$$

– very easy to implement so you estimate Δ at the same time as estimating the price V

Likelihood ratio method

Similarly for vega we have

$$\frac{\partial \log p}{\partial \sigma} = -\frac{1}{\sigma} - \frac{\log(S/S_0) - (r - \frac{1}{2}\sigma^2)T}{\sigma} + \frac{\left(\log(S/S_0) - (r - \frac{1}{2}\sigma^2)T\right)^2}{\sigma^3 T}$$

and hence

$$\text{vega} = \mathbb{E}\left[\left(\frac{1}{\sigma}\left(\frac{W(T)^2}{T} - 1\right) - W(T)\right) f(S_T)\right]$$

Likelihood ratio method

In both cases, the variance is very large when σ is small, and it is also large for Δ when T is small.

More generally, LRM is usually the approach with the largest variance.

Likelihood ratio method

To get second derivatives, note that

$$\frac{\partial^2 \log p}{\partial \theta^2} = \frac{\partial}{\partial \theta}\left(\frac{1}{p}\,\frac{\partial p}{\partial \theta}\right) = \frac{1}{p}\,\frac{\partial^2 p}{\partial \theta^2} - \frac{1}{p^2}\left(\frac{\partial p}{\partial \theta}\right)^2$$

$$\Longrightarrow \quad \frac{1}{p}\,\frac{\partial^2 p}{\partial \theta^2} = \frac{\partial^2 \log p}{\partial \theta^2} + \left(\frac{\partial \log p}{\partial \theta}\right)^2$$

and hence

$$\frac{\partial^2 V}{\partial \theta^2} = \mathbb{E}\left[\left(\frac{\partial^2 \log p}{\partial \theta^2} + \left(\frac{\partial \log p}{\partial \theta}\right)^2\right) f(S_T)\right]$$

Likelihood ratio method

In the multivariate extension, $X = \log S_T$ can be written as

$$X = \mu + L\, Z$$

where μ is the mean vector, $\Sigma = L\, L^T$ is the covariance matrix and Z is a vector of uncorrelated Normals. The joint p.d.f. is

$$\log p = -\tfrac{1}{2} \log |\Sigma| - \tfrac{1}{2} (X - \mu)^T \Sigma^{-1} (X - \mu) - \tfrac{1}{2} d\, \log(2\pi).$$

and after a lot of algebra we obtain

$$\frac{\partial \log p}{\partial \mu} = L^{-T} Z,$$

$$\frac{\partial \log p}{\partial \Sigma} = \tfrac{1}{2}\, L^{-T} \left(Z Z^T - I \right) L^{-1}$$

Likelihood Ratio Method

Extending this to a SDE path simulation with M timesteps, with the payoff a function purely of the discrete states $\widehat{S}_n$, we have the M-dimensional integral

$$V = \mathbb{E}[\widehat{f}(\widehat{S})] = \int \widehat{f}(\widehat{S})\, p(\widehat{S})\, \mathrm{d}\widehat{S},$$

where $$\mathrm{d}\widehat{S} \equiv \mathrm{d}\widehat{S}_1\ \mathrm{d}\widehat{S}_2\ \mathrm{d}\widehat{S}_3\ \ldots\ \mathrm{d}\widehat{S}_M$$

and $p(\widehat{S})$ is the product of the p.d.f.s for each timestep

$$\begin{aligned} p(\widehat{S}) &= \prod_n p_n(\widehat{S}_{n+1}|\widehat{S}_n) \\ \log p(\widehat{S}) &= \sum_n \log p_n(\widehat{S}_{n+1}|\widehat{S}_n) \end{aligned}$$

Likelihood Ratio Method

For the Euler-Maruyama approximation of Geometric Brownian Motion,

$$\log p_n = -\log \widehat{S}_n - \log \sigma - \tfrac{1}{2}\log(2\pi h) - \tfrac{1}{2}\,\frac{\left(\widehat{S}_{n+1} - \widehat{S}_n(1+r\,h)\right)^2}{\sigma^2\, \widehat{S}_n^2\, h}$$

$$\begin{aligned}\Longrightarrow \quad \frac{\partial(\log p_n)}{\partial\sigma} &= -\frac{1}{\sigma} + \frac{\left(\widehat{S}_{n+1} - \widehat{S}_n(1+r\,h)\right)^2}{\sigma^3\, \widehat{S}_n^2\, h} \\ &= \frac{Z_n^2 - 1}{\sigma}\end{aligned}$$

where Z_n is the unit Normal defined by

$$\widehat{S}_{n+1} - \widehat{S}_n(1+r\,h) = \sigma\, \widehat{S}_n\, \sqrt{h}\, Z_n$$

Likelihood Ratio Method

Hence, the approximation of Vega is

$$\frac{\partial}{\partial \sigma}\, \mathbb{E}[f(\widehat{S}_M)] = \mathbb{E}\left[\left(\sum_n \frac{Z_n^2 - 1}{\sigma} \right) f(\widehat{S}_M) \right]$$

Note that again this gives zero for $f(S) \equiv 1$.

Note also that $\mathbb{V}[Z_n^2 - 1] = 2$ and therefore

$$\mathbb{V}\left[\left(\sum_n \frac{Z_n^2 - 1}{\sigma} \right) f(\widehat{S}_M) \right] \;=\; O(M) \;=\; O(T/h)$$

This $O(h^{-1})$ blow-up is the great drawback of the LRM.

Pathwise sensitivities

Start instead with

$$V \equiv \mathbb{E}\left[f(S_T)\right] = \int f(S_T)\, p_W(W)\, \mathrm{d}W$$

and differentiate this to get

$$\frac{\partial V}{\partial \theta} = \int \frac{\partial f}{\partial S}\, \frac{\partial S_T}{\partial \theta}\, p_W\, \mathrm{d}W = \mathbb{E}\left[\frac{\partial f}{\partial S}\, \frac{\partial S_T}{\partial \theta}\right]$$

with $\partial S_T/\partial\theta$ being evaluated at fixed W.

Note: this derivation needs $f(S)$ to be differentiable, but by considering the limit of a sequence of smoothed (regularised) functions can prove it's OK provided $f(S)$ is continuous and piecewise differentiable

Pathwise sensitivities

This leads to the estimator

$$\frac{1}{N} \sum_{i=1}^{N} \frac{\partial f}{\partial S}(S^{(i)}) \; \frac{\partial S^{(i)}}{\partial \theta}$$

which is the derivative of the usual price estimator

$$\frac{1}{N} \sum_{i=1}^{N} f(S^{(i)})$$

Gives incorrect estimates when $f(S)$ is discontinuous.
e.g. for digital put $\dfrac{\partial f}{\partial S} = 0$ so estimated value of Greek is zero – clearly wrong.

Pathwise sensitivities

Extension to second derivatives is straightforward

$$\begin{aligned}
\frac{\partial^2 V}{\partial \theta^2} &= \int \left\{ \frac{\partial^2 f}{\partial S^2} \left(\frac{\partial S_T}{\partial \theta} \right)^2 + \frac{\partial f}{\partial S}\, \frac{\partial^2 S_T}{\partial \theta^2} \right\} p_W \; \mathrm{d}W \\
&= \mathbb{E}\left[\frac{\partial^2 f}{\partial S^2} \left(\frac{\partial S_T}{\partial \theta} \right)^2 + \frac{\partial f}{\partial S}\, \frac{\partial^2 S_T}{\partial \theta^2} \right]
\end{aligned}$$

with $\partial^2 S_T / \partial \theta^2$ also being evaluated at fixed W.

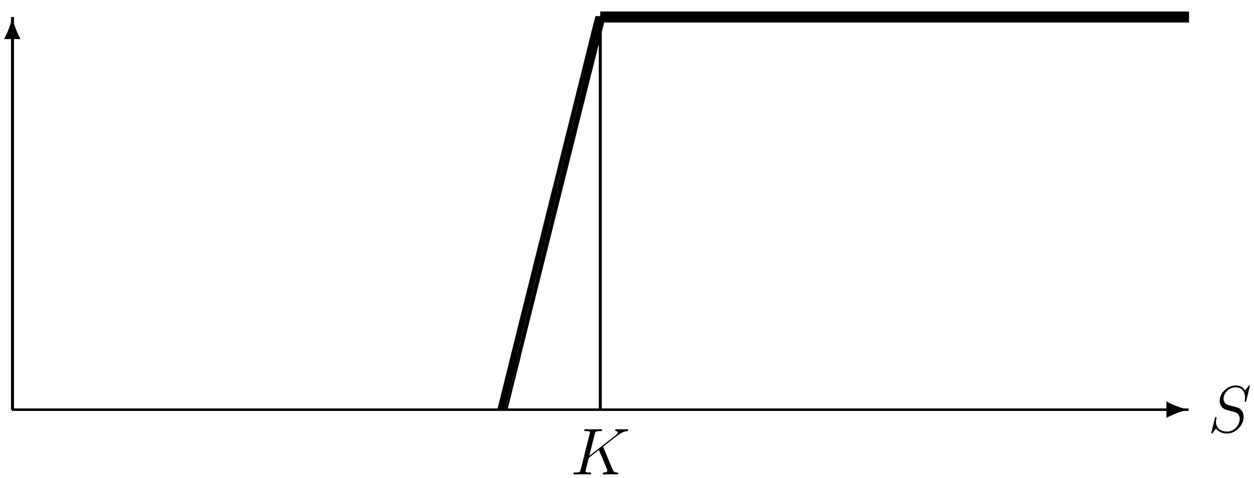
However, this requires $f(S)$ to have a continuous first derivative – a problem in practice

Pathwise sensitivities

To handle payoffs which do not have the necessary continuity/smoothness one can modify the payoff

For digital options it is common to use a piecewise linear approximation to limit the magnitude of Δ near maturity – avoids large transaction costs

Bank selling the option will price it conservatively (i.e. over-estimate the price)



Pathwise sensitivities

The standard call option definition can be smoothed by integrating the smoothed Heaviside function

$$H_\varepsilon(S\!-\!K) = \Phi\left(\frac{S\!-\!K}{\varepsilon}\right)$$

with $\varepsilon \ll K$, to get

$$f(S) = (S\!-\!K)\ \Phi\left(\frac{S\!-\!K}{\varepsilon}\right) + \frac{\varepsilon}{\sqrt{2\pi}} \exp\left(-\ \frac{(S\!-\!K)^2}{2\,\varepsilon^2}\right)$$

This will allow the calculation of Γ and other second derivatives

Pathwise Sensitivity

To allow for possibility of calculating sensitivity to changes in correlation, better to start with integral with respect to unit Normal Z:

$$V = \mathbb{E}[f(S_T)] = \int f(S_T)\, \phi(Z)\, \mathrm{d}Z$$

where $\phi(Z)$ is unit Normal p.d.f.

Differentiation then gives

$$\frac{\partial V}{\partial \theta} = \mathbb{E}\left[\frac{\partial f}{\partial S}\, \frac{\partial S_T}{\partial \theta} \right]$$

with $\partial S_T / \partial \theta$ being evaluated at fixed Z.

Pathwise Sensitivity

In the multiple dimensional GBM case,

$$S_i(T) = S_i(0)\exp((r - \tfrac{1}{2}\sigma_i^2)T + \sigma_i\sqrt{T}\,(LZ)_i)$$

where $L\,L^T$ is the correlation matrix for $\mathrm{d}W$, and the components of Z are i.i.d. unit Normals.

Hence for vega, we have

$$\left.\frac{\partial S_i}{\partial \sigma_i}\right|_Z = S_i(T)\left(-\,\sigma_i T + \sqrt{T}\,(LZ)_i\right)$$

Pathwise Sensitivity

The extension to SDE path simulations is quite natural, with

$$V = \mathbb{E}[\widehat{f}(\widehat{S})] = \int \widehat{f}(\widehat{S}(Z))\, \phi(Z)\, \mathrm{d}Z$$

where $\mathrm{d}Z \equiv \mathrm{d}Z_0\, \mathrm{d}Z_1\, \mathrm{d}Z_2\, \ldots\, \mathrm{d}Z_{M-1}$ and $\phi(Z)$ is the product of the unit Normal p.d.f.'s $\phi(Z) = \prod_n \phi(Z_n)$

Differentiation then gives

$$\frac{\partial V}{\partial \theta} = \mathbb{E}\left[\frac{\partial \widehat{f}}{\partial \widehat{S}}\, \frac{\partial \widehat{S}}{\partial \theta}\right] \equiv \mathbb{E}\left[\sum_m \frac{\partial \widehat{f}}{\partial \widehat{S}_m}\, \frac{\partial \widehat{S}_m}{\partial \theta}\right]$$

with $\partial \widehat{S}/\partial \theta$ being evaluated at fixed Z.

Pathwise Sensitivity

For a scalar GBM, defining $\widehat{s}_n \equiv \dfrac{\partial \widehat{S}_n}{\partial \sigma}$ then differentiating the initial data $\widehat{S}_0 = S(0)$ gives $\widehat{s}_0 = 0$, and differentiating

$$\widehat{S}_{n+1} = \widehat{S}_n \left(1 + r\,h + \sigma\,\sqrt{h}\,Z_n\right)$$

gives

$$\widehat{s}_{n+1} = \widehat{s}_n \left(1 + r\,h + \sigma\,\sqrt{h}\,Z_n\right) + \widehat{S}_n\,\sqrt{h}\,Z_n$$

and then

$$\text{Vega} = \mathbb{E}\left[\frac{\partial \widehat{f}}{\partial \widehat{S}}\,\widehat{s}\right] \equiv \mathbb{E}\left[\sum_m \frac{\partial \widehat{f}}{\partial \widehat{S}_m}\,\widehat{s}_m\right]$$

Pathwise Sensitivity

As $h \to 0$,

$$\widehat{s}_m \to \frac{\partial S_t}{\partial \sigma}$$

so the approximate path sensitivity tends to the true value, and hence both the expectation and variance of

$$\frac{\partial \widehat{f}}{\partial \widehat{S}} \, \widehat{s}$$

converge to the expectation and variance of

$$\frac{\partial f}{\partial \sigma}$$

Thus, there is no variance "blow-up".

Final Words

Estimating Greeks is an important task:

- LRM can handle discontinuous payoffs, but a little complicated for multivariate case, and the variance blows up as $h \to 0$
- pathwise sensitivity is usually the best approach (simplest, lowest variance and least cost) when it is applicable – needs continuous payoff for first derivatives
- payoff smoothing can be used to make pathwise approach applicable to discontinuous payoffs and for second derivatives
- alternatively, combine pathwise sensitivity with finite differences for second derivatives – e.g. use pathwise to compute Δ, then finite difference this to get Γ

Early Exercise

Perhaps the biggest challenge for Monte Carlo methods is the accurate and efficient pricing of options with optional early exercise:

- Bermudan options: can exercise at a finite number of times t_j
- American options: can exercise at any time

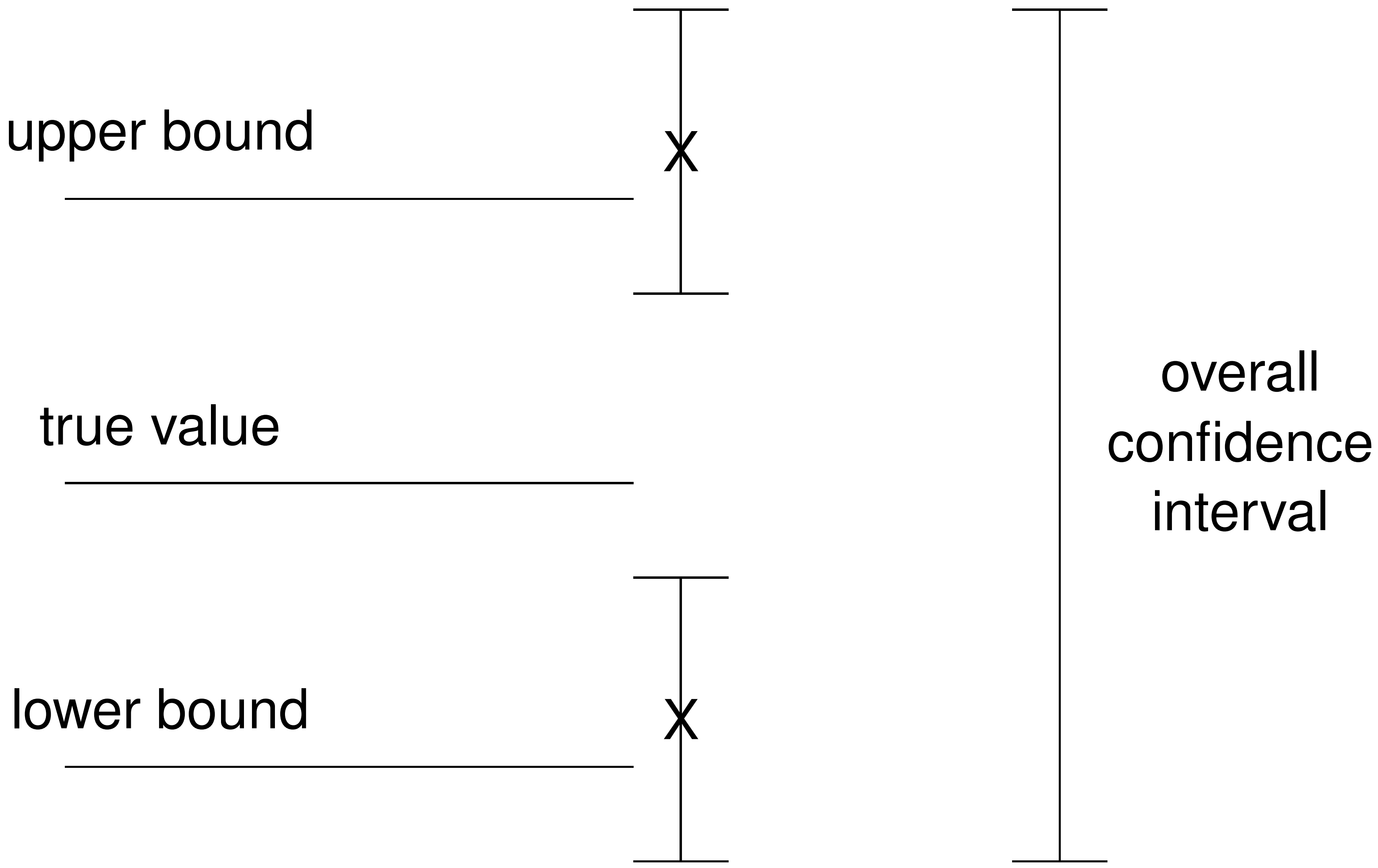
The challenge is to find/approximate the optimal strategy (i.e. when to exercise) and hence determine the price and Greeks.

Early Exercise

Approximating the optimal exercise boundary introduces new approximation errors:

- an approximate exercise boundary is inevitably sub-optimal
  $\Longrightarrow$ under-estimate of "true" value, but accurate value for the sub-optimal strategy
- for the option buyer, sub-optimal price reflects value achievable with sub-optimal strategy
- for the option seller, "true" price is best a purchaser might achieve
- can also derive an upper bound approximation

Early Exercise



Early Exercise

Why is early exercise so difficult for Monte Carlo methods?

- leads naturally to a dynamic programming formulation working backwards in time
- fairly minor extension for finite difference methods which already march backwards in time
- doesn't fit well with Monte Carlo methods which go forwards in time

Problem Formulation

Following description in Glasserman's book, "Monte Carlo Methods in Financial Engineering" the Bermudan problem has the dynamic programming formulation:

$$
\begin{aligned}
\widetilde{V}_m(x) &= \widetilde{h}_m(x) \\
\widetilde{V}_{i-1}(x) &= \max\left(\widetilde{h}_{i-1}(x), \mathbb{E}[D_{i-1,i}\,\widetilde{V}_i(X_i) \mid X_{i-1} = x]\right)
\end{aligned}
$$

where

- X_i is the underlying at exercise time t_i
- $\widetilde{V}_i(x)$ is option value at time t_i assuming not previously exercised
- $\widetilde{h}_i(x)$ is exercise value at time t_i
- $D_{i-1,i}$ is the discount factor for interval $[t_{i-1}, t_i]$

Problem Formulation

By defining

$$
\begin{aligned}
h_i(x) &= D_{0,i}\, \widetilde{h}_i(x) \\
V_i(x) &= D_{0,i}\, \widetilde{V}_i(x)
\end{aligned}
$$

where

$$D_{0,i} = D_{0,1}\ D_{1,2}\ \ldots\ D_{i-1,i}$$

can simplify the formulation to

$$
\begin{aligned}
V_m(x) &= h_m(x) \\
V_{i-1}(x) &= \max\left(h_{i-1}(x), \mathbb{E}[V_i(X_i) \mid X_{i-1} = x]\right)
\end{aligned}
$$

Problem Formulation

An alternative point of view considers stopping rules τ, the time at which the option is exercised.

For a particular stopping rule, the initial option value is

$$V_0(X_0) = \mathbb{E}[h_\tau(X_\tau)],$$

the expected value of the option at the time of exercise.

The best that can be achieved is then

$$V_0(X_0) = \sup_\tau \mathbb{E}[h_\tau(X_\tau)]$$

giving an optimisation problem.

Problem Formulation

The continuation value is

$$C_i(x) = \mathbb{E}[V_{i+1}(X_{i+1}) \mid X_i = x]$$

and so the optimal stopping rule is

$$\tau = \min\{i : h_i(X_i) > C_i(X_i)\}$$

Approximating the continuation value leads to an approximate stopping rule.

Longstaff-Schwartz Method

The Longstaff-Schwartz method (2001) is the one most used in practice.

Start with N path simulations, each going from initial time $t=0$ to maturity $t=T=t_m$.

Problem is to assign a value to each path, working out whether and when to exercise the option.

This is done by working backwards in time, approximating the continuation value.

Longstaff-Schwartz Method

At maturity, the value of an option is

$$V_m(X_m) = h_m(X_m)$$

At the previous exercise date, the continuation value is

$$C_{m-1}(x) = \mathbb{E}[V_m(X_m) \mid X_{m-1} = x]$$

This is approximated using a set of R basis functions as

$$\widehat{C}_{m-1}(x) = \sum_{r=1}^{R} \beta_r \, \psi_r(x)$$

Longstaff-Schwartz Method

The coefficients β_r are obtained by a least-squares minimisation, minimising

$$\mathbb{E}\left\{ \left(\mathbb{E}[V_m(X_m) \mid X_{m-1}] - \widehat{C}_{m-1}(X_{m-1}) \right)^2 \right\}$$

Setting the derivative w.r.t. β_r to zero gives

$$\mathbb{E}\left\{ \left(\mathbb{E}[V_m(X_m) \mid X_{m-1}] - \widehat{C}_{m-1}(X_{m-1}) \right) \psi_r(X_{m-1}) \right\} = 0$$

and hence

$$\begin{aligned} \mathbb{E}[V_m(X_m)\, \psi_r(X_{m-1})] &= \mathbb{E}[\widehat{C}_{m-1}(X_{m-1})\, \psi_r(X_{m-1})] \\ &= \sum_s \mathbb{E}[\psi_r(X_{m-1})\, \psi_s(X_{m-1})]\, \beta_s \end{aligned}$$

Longstaff-Schwartz Method

This set of equations can be written collectively as

$$B_{\psi\psi}\,\beta = B_{V\psi}$$

where

$$(B_{V\psi})_r = \mathbb{E}[V_m(X_m)\,\psi_r(X_{m-1})]$$

$$(B_{\psi\psi})_{rs} = \mathbb{E}[\psi_r(X_{m-1})\,\psi_s(X_{m-1})]$$

Therefore,

$$\beta = B_{\psi\psi}^{-1}\,B_{V\psi}$$

Longstaff-Schwartz Method

In the numerical approximation, each of the expectations is replaced by an average over the values from the N paths.

For example,

$$\mathbb{E}[\psi_r(X_{m-1})\,\psi_s(X_{m-1})]$$

is approximated as

$$N^{-1}\sum_{n=1}^{N}\psi_r(X_{m-1}^{(n)})\ \psi_s(X_{m-1}^{(n)})$$

Assuming that the number of paths is much greater than the number of basis functions, the main cost is in approximating $B_{\psi\psi}$ with a cost which is $O(N\,R^2)$.

Longstaff-Schwartz Method

Once we have the approximation for the continuation value, what do we do?

- if $\widehat{C}(X_{m-1}) < h_{m-1}(X_{m-1})$, exercise the option and set

$$V_{m-1} = h_{m-1}(X_{m-1})$$

- if not, then set

$$V_{m-1} = \widehat{C}(X_{m-1})$$

  (Tsitsiklis & van Roy, 1999), <u>or</u>

$$V_{m-1} = V_m$$

  (Longstaff & Schwartz, 2001)

Longstaff-Schwartz Method

The Longstaff-Schwartz treatment only uses the continuation estimate to decide on the exercise boundary

- no loss of accuracy for paths which are not exercised
  - reduces to standard Monte Carlo estimate for European option if there is no early exercise

- as an extra bonus, only need to estimate $\widehat{C}(x)$ for paths which are in-the-money (i.e. $h(x) > 0$)
  - approximating $\widehat{C}(x)$ for a smaller range of x can be done more accurately for a given number of basis functions

Longstaff-Schwartz Method

Provided the basis functions are chosen suitably, the approximation

$$\widehat{C}_{m-1}(x) = \sum_{r=1}^{R} \beta_r \, \psi_r(x)$$

gets increasingly accurate as $R \to \infty$ – Longstaff & Schwartz used 5-20 basis functions in their paper.

Having completed the calculation for t_{m-1}, repeat the procedure for t_{m-2} then t_{m-3} and so on. Could use different basis functions for each exercise time – the coefficients β will certainly be different.

Longstaff-Schwartz Method

The estimate will tend to be biased low because of the sub-optimal exercise boundary, however might be biased high due to using the same paths for decision-making and valuation.

To be sure of being biased low, should use two sets of paths, one to estimate the continuation value and exercise boundary, and the other for the valuation.

This leaves the problem of computing an upper bound.

Upper Bounds

In Glasserman's Bermudan version of Roger's continuous time result (2002), he lets M_m be a martingale with $M_0 = 0$.

For any stopping rule τ, we have

$$\mathbb{E}[h_\tau(X_\tau)] \ = \ \mathbb{E}[h_\tau(X_\tau) - M_\tau] \ \leq \ \mathbb{E}[\max_k (h_k(X_k) - M_k)]$$

This is true for all martingales M and all stopping rules τ and hence

$$V_0(X_0) \ = \ \sup_\tau \mathbb{E}[h_\tau(X_\tau)] \ \leq \ \inf_M \mathbb{E}[\max_k (h_k(X_k) - M_k)]$$

Upper Bounds

The key duality result is that in fact there is equality

$$\sup_\tau \mathbb{E}[h_\tau(X_\tau)] \;=\; \inf_M \mathbb{E}[\max_k (h_k(X_k) - M_k)]$$

so that

- an arbitrary τ gives a lower bound
- an arbitrary M gives an upper bound
- making both of them “better” shrinks the gap between them to zero

Upper Bounds

Glasserman proves by induction that the optimal martingale M is equal to

$$M_k = \sum_1^k \Big(V_i(X_i) - \mathbb{E}[V_i(X_i) \mid X_{i-1}] \Big)$$

To get a good upper bound we approximate this martingale.

Upper Bounds

The approximate martingale for a particular path is defined as

$$\widehat{M}_k = \sum_1^k \left(V_i(X_i) - P^{-1} \sum_p V_i(X_i^{(p)}) \right)$$

where the $X_i^{(p)}$ are values for X_i from P different mini-paths starting at X_{i-1}, and

$$V_i(X_i) = \max(h_i(X_i), \widehat{C}_i(X_i))$$

with $\widehat{C}_i(X_i)$ being the approximate continuation value given by the Longstaff-Schwartz algorithm.

Glasserman suggests up to 100 mini-paths may be needed.

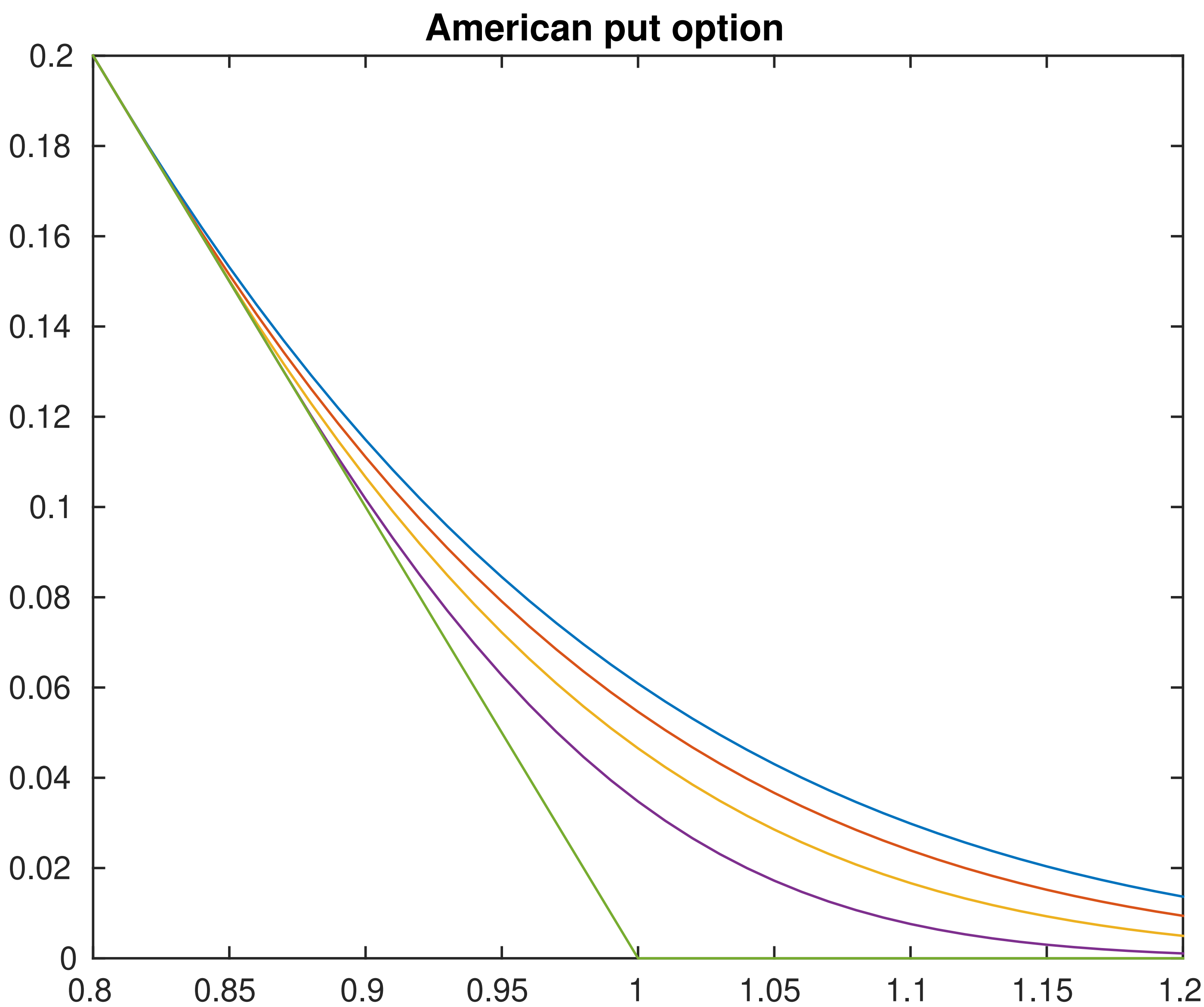
Numerical results

- Single asset, Geometric Brownian Motion: $r = 0.05$, $\sigma = 0.2$, $T = 1$
- American put option: $S_0 = 1$, $K = 1$
- explicit finite difference method used to give "true value"
- Longstaff-Schwartz method uses 64 timesteps, 10^5 paths, 3 basis functions: 1, $x-1$, $(x-1)^2$

Computed values:

method	value	err. bnd
finite difference method	0.0609	???
Longstaff-Schwartz, first set	0.0605	$\pm$ 0.0006
Longstaff-Schwartz, second set	0.0609	$\pm$ 0.0006

Numerical results



Final Words

- Bermudan and American options are important applications
- Longstaff-Schwartz method is popular, but maybe still scope for improvement
- finite difference method (bumping) is probably used for Greeks
- is second independent set of paths used in practice?
- are upper bounds used in practice?