Monte Carlo Methods

Prof. Mike Giles

mike.giles@maths.ox.ac.uk

Oxford University Mathematical Institute

Monte Carlo methods

In option pricing there are two main approaches:

- Monte Carlo methods for estimating expected values of financial payoff functions based on underlying assets.

  E.g., we want to estimate $\mathbb{E}[f(S(T))]$ where

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

  and $W(T)$ is driving Brownian motion at terminal time T

- Numerical approximation of the PDE which describes the evolution of the expected value.

  $$u(s,t) = \mathbb{E}\left[f(S(T)) \mid S(t)=s\right]$$

  Usually less costly than MC when there are very few underlying assets ($M \leq 3$), but much more expensive when there are many.

Geometric Brownian Motion

In this first lecture, we consider M underlying assets, each modelled by Geometric Brownian Motion

$$\mathrm{d}S_i = r\, S_i\, \mathrm{d}t + \sigma_i\, S_i\, \mathrm{d}W_i$$

so Ito calculus gives us

$$S_i(T) = S_i(0) \ \exp\left((r - \tfrac{1}{2}\sigma_i^2)T + \sigma_i W_i(T)\right)$$

in which each $W_i(T)$ is Normally distributed with zero mean and variance T.

We can use standard Random Number Generation software (e.g. `randn` function in MATLAB) to generate samples of each $W_i(T)$, but there is a problem ...

Correlated Brownian Motions

Different assets do not behave independently – on average, they tend to move up and down together.

This is modelled by introducing correlation between the driving Brownian motions so that

$$\mathbb{E}\left[W_i(T)\ W_j(T)\right] = \Omega_{i,j}\, T$$

where $\Omega_{i,j}$ is the correlation coefficient, and hence

$$\mathbb{E}\left[W(T)\ W(T)^T\right] = \Omega\, T.$$

How do we generate samples of $W_i(T)$?

Correlated Normal Random Variables

Suppose x is a vector of independent $N(0,1)$ variables, and define a new vector $y = L\,x$.

Each element of y is Normally distributed, $\mathbb{E}[y] = L\,\mathbb{E}[x] = 0$, and

$$\mathbb{E}[y\,y^T] = \mathbb{E}[L\,x\,x^T\,L^T] = L\,\mathbb{E}[x\,x^T]\,L^T = L\,L^T.$$

since $\mathbb{E}[x\,x^T] = I$ because

- elements of x are independent $\Longrightarrow \mathbb{E}[x_i\,x_j] = 0$ for $i \neq j$
- elements of x have unit variance $\Longrightarrow \mathbb{E}[x_i^2] = 1$

Correlated Normal Random Variables

To get $\mathbb{E}[y\,y^T] = \Omega$, we need to find L such that

$$L\,L^T = \Omega$$

L is not uniquely defined – simplest choice is to use a Cholesky factorization in which L is lower-triangular, with a positive diagonal.

In MATLAB, this is achieved using

```
L = chol(Omega,'lower');
```


Basket Call Option

In a basket call option, the discounted payoff function is

$$f = \exp(-r\,T)\,\left(\frac{1}{M}\sum_i S_i(T) - K\right)^+$$

where K is the strike, r is the risk-free interest rate, and $(x)^+ \equiv \max(x,0)$.

Monte Carlo estimation of $\mathbb{E}[f(S(T))]$ is very simple – we generate N independent samples of $W(T)$, compute $S(T)$, and then average to get

$$\overline{f}_N \equiv N^{-1}\sum_{n=1}^{N} f^{(n)} \;\approx\; \mathbb{E}[f]$$

Monte Carlo Error and CLT

This MC estimate is unbiased, meaning that the average error is zero

$$\mathbb{E}[\varepsilon_N] = 0$$

where ε_N is the error $\overline{f}_N - \mathbb{E}[f]$.

In addition, the Central Limit Theorem proves that for large N the error is asymptotically Normally distributed

$$\varepsilon_N(f) \sim \sigma\,N^{-1/2}\,Z$$

with Z a $N(0,1)$ random variable and σ^2 the variance of f:

$$\sigma^2 = \mathbb{V}[f] \equiv \mathbb{E}\left[(f-\mathbb{E}[f])^2\right].$$

CLT

This means that

$$\mathbb{P}\left[\left|N^{1/2}\sigma^{-1}\varepsilon_N\right| < s\right] \;\approx\; 1 - 2\,\Phi(-s),$$

where $\Phi(s)$ is the Normal CDF (cumulative distribution function).

Typically we use $s=3$, corrsponding to a 3-standard deviation confidence interval, with $1-2\,\Phi(-s)\approx 0.997$.

Hence, with probability 99.7%, we have

$$\left|N^{1/2}\sigma^{-1}\varepsilon_N\right| < 3 \;\;\Longrightarrow\;\; |\varepsilon_N| < 3\,\sigma\,N^{-1/2}$$

This bounds the accuracy, but we need an estimate for σ.

Empirical Variance

Given N samples, the empirical variance is

$$\widetilde{\sigma}^2 = N^{-1}\sum_{n=1}^{N}\left(f^{(n)} - \overline{f}_N\right)^2 = \overline{f^2} - (\overline{f})^2$$

where

$$\overline{f} = N^{-1}\sum_{n=1}^{N} f^{(n)}, \qquad \overline{f^2} = N^{-1}\sum_{n=1}^{N}\left(f^{(n)}\right)^2$$

$\widetilde{\sigma}^2$ is a slightly biased estimator for σ^2 – an unbiased estimator is

$$\widehat{\sigma}^2 = \frac{N}{N\!-\!1}\,\widetilde{\sigma}^2 = \frac{N}{N\!-\!1}\left(\overline{f^2} - (\overline{f})^2\right)$$

Applications

Geometric Brownian motion for single asset:

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

For the European call option,

$$f(S) = \exp(-rT)\;(S\!-\!K)^+$$

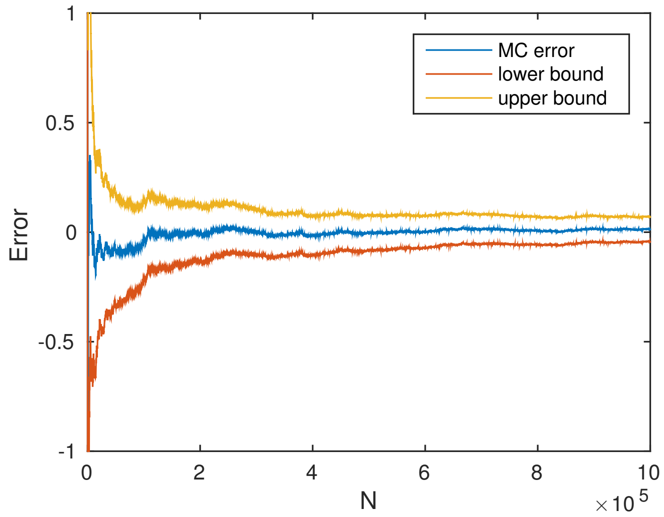
where K is the strike price.

For numerical experiments we will consider a European call with $\;r\!=\!0.05,\;\sigma=0.2,\;T\!=\!1,\;S_0\!=\!110,\;K\!=\!100.$

The analytic value is known for comparison.

Applications

MC calculation with up to 10^6 paths; true value = 17.663



Applications

The upper and lower bounds are given by

$$\text{Mean} \pm \frac{3\,\widetilde{\sigma}}{\sqrt{N}},$$

so more than a 99.7% probability that the true value lies within these bounds.

Applications

MATLAB code:

```
r=0.05;  sig=0.2;  T=1;  S0=110;  K=100;
N = 1:1000000;
Y = randn(1,max(N));   % Normal random variables
S = S0*exp((r-sig^2/2)*T + sig*sqrt(T)*Y);
F = exp(-r*T)*max(0,S-K);

sum1 = cumsum(F);      % cumulative summation of
sum2 = cumsum(F.^2);   % payoff and its square
val  = sum1./N;
sd   = sqrt(sum2./N - val.^2);
```

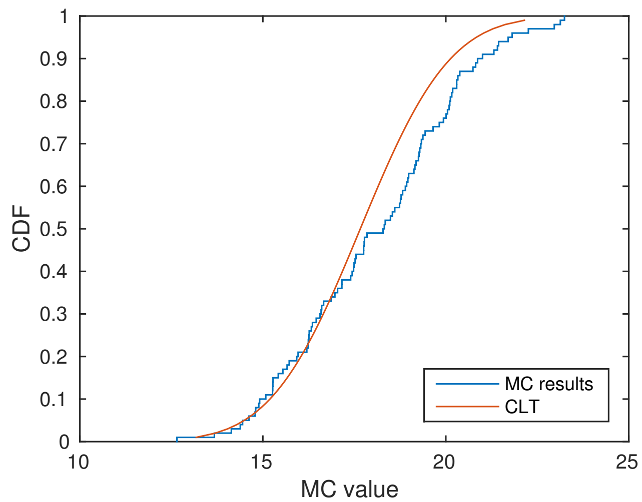

Applications

```
err = european_call(r,sig,T,S0,K,'value') - val;

plot(N,err,                    ...
     N,err-3*sd./sqrt(N),      ...
     N,err+3*sd./sqrt(N))
axis([0 length(N) -1 1])
xlabel('N'); ylabel('Error')
legend('MC error','lower bound','upper bound')
```

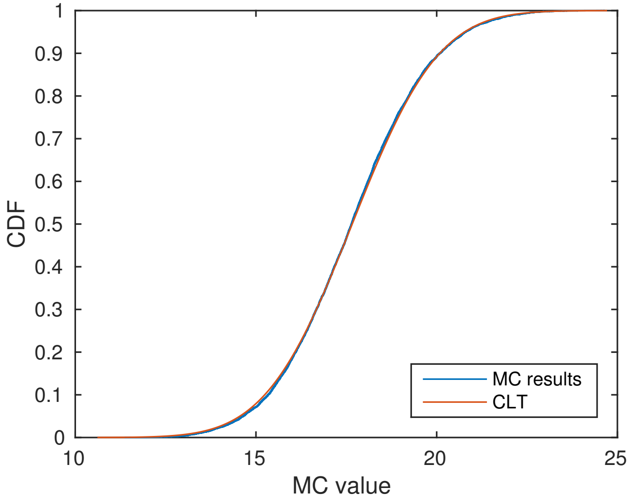

Applications

CLT: 100 independent tests, each with 100 samples



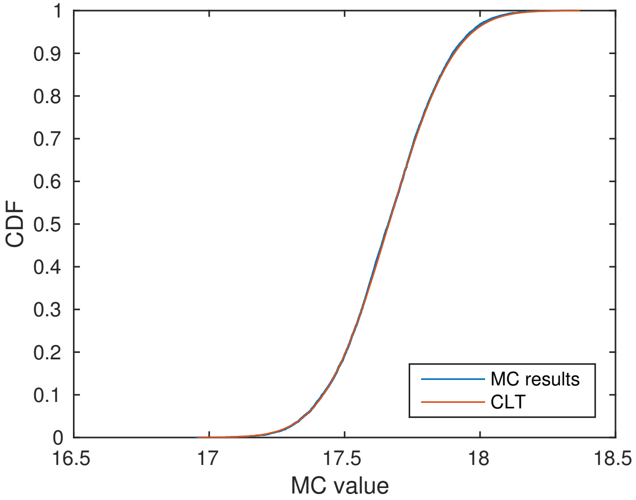
Applications

CLT: 10^4 independent tests, each with 100 samples



Applications

CLT: 10^4 independent tests, each with 10^4 samples



Basket call option

- 5 underlying assets starting at $S_0 = 100$, with call option on arithmetic mean with strike $K = 100$
- Geometric Brownian Motion model, $r = 0.05, T = 1$
- volatility $\sigma = 0.2$ and correlation matrix

$$\Omega = \begin{pmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{pmatrix}$$

Applications

```
r=0.05; sigma=0.2; rho=0.1; T=1; K=100; S0=100;

N = 10^5;                     % number of MC samples
Omega = eye(5) + rho*(ones(5)-eye(5));
L = chol(Omega,'lower');   % Cholesky factorisation
W = sqrt(T)*L*randn(5,N);
S = S0.*exp((r-0.5*sigma^2)*T + sigma*W);

S = 0.2*sum(S,1);             % average asset value
F = exp(-r*T)*max(S-K,0); % call option

val = sum(F)/N                % mean and its std. dev.
sd  = sqrt( (sum(F.^2)/N - val.^2)/(N-1) )
```

Variance Reduction

Monte Carlo is a very simple method; it gets complicated when we try to reduce the variance, and hence the number of samples required.

There are several approaches:

- antithetic variables
- control variates
- importance sampling
- stratified sampling
- Latin hypercube
- quasi-Monte Carlo

We will discuss control variates.

Review of elementary results

If a, b are random variables, and λ, μ are constants, then

$$\begin{aligned}
\mathbb{E}[a+\mu] &= \mathbb{E}[a]+\mu \\
\mathbb{V}[a+\mu] &= \mathbb{V}[a] \\
\mathbb{E}[\lambda\, a] &= \lambda\, \mathbb{E}[a] \\
\mathbb{V}[\lambda\, a] &= \lambda^2\, \mathbb{V}[a] \\
\mathbb{E}[a+b] &= \mathbb{E}[a]+\mathbb{E}[b] \\
\mathbb{V}[a+b] &= \mathbb{V}[a] + 2\,\text{Cov}[a,b] + \mathbb{V}[b]
\end{aligned}$$

where

$$\begin{aligned}
\mathbb{V}[a] &\equiv \mathbb{E}\left[(a-\mathbb{E}[a])^2\right] = \mathbb{E}\left[a^2\right] - (\mathbb{E}[a])^2 \\
\text{Cov}[a,b] &\equiv \mathbb{E}\left[(a-\mathbb{E}[a])\,(b-\mathbb{E}[b])\right]
\end{aligned}$$

Review of elementary results

If a, b are independent random variables then

$$\mathbb{E}[f(a)\; g(b)] = \mathbb{E}[f(a)]\; \mathbb{E}[g(b)]$$

Hence, $\text{Cov}[a,b]=0$ and therefore $\mathbb{V}[a+b] = \mathbb{V}[a]+\mathbb{V}[b]$

Extending this to a set of N iid (independent identically distributed) r.v.'s x_n, we have

$$\mathbb{V}\left[\sum_{n=1}^N x_n\right] = \sum_{n=1}^N \mathbb{V}[x_n] = N\, \mathbb{V}[x]$$

and so

$$\mathbb{V}\left[N^{-1}\sum_{n=1}^N x_n\right] = N^{-1}\, \mathbb{V}[x]$$

Control Variates

Suppose we want to approximate $\mathbb{E}[f]$ using a simple Monte Carlo average $\overline{f}$.

If there is another payoff g for which we know $\mathbb{E}[g]$, can use $\overline{g}-\mathbb{E}[g]$ to reduce error in $\overline{f}-\mathbb{E}[f]$.

How? By defining a new estimator

$$\widehat{f} = \overline{f} - \lambda\,(\overline{g}-\mathbb{E}[g])$$

Again unbiased since $\mathbb{E}[\widehat{f}] = \mathbb{E}[\overline{f}] = \mathbb{E}[f]$

Control Variates

For a single sample,

$$\mathbb{V}[f - \lambda\,(g - \mathbb{E}[g])] = \mathbb{V}[f] - 2\,\lambda\,\text{Cov}[f,g] + \lambda^2\,\mathbb{V}[g]$$

For an average of N samples,

$$\mathbb{V}[\overline{f} - \lambda\,(\overline{g} - \mathbb{E}[g])] = N^{-1}\left(\mathbb{V}[f] - 2\,\lambda\,\text{Cov}[f,g] + \lambda^2\,\mathbb{V}[g]\right)$$

To minimise this, the optimum value for λ is

$$\lambda = \frac{\text{Cov}[f,g]}{\mathbb{V}[g]}$$

Control Variates

The resulting variance is

$$N^{-1}\,\mathbb{V}[f]\left(1 - \frac{(\text{Cov}[f,g])^2}{\mathbb{V}[f]\,\mathbb{V}[g]}\right) = N^{-1}\,\mathbb{V}[f]\,\left(1-\rho^2\right)$$

where ρ is the correlation between f and g.

The challenge is to choose a good g which is well correlated with f – the covariance, and hence the optimal λ, can be estimated from the data.

Control Variates

For a basket option with M underlying assets, we know that for each asset

$$\mathbb{E}[\exp(-r\,T)\,S_i(T)] = S_i(0)$$

so we could use

$$g = \exp(-r\,T)\,\frac{1}{M}\sum_{m=1}^{M} S_i(T)$$

with

$$\mathbb{E}[g] = \frac{1}{M}\sum_{m=1}^{M} S_i(0)$$

Numerical test will do the simpler scalar case with $M\!=\!1$.

Application

MATLAB code, part 1 – estimating optimal λ:

```
r=0.05; sig=0.2; T=1; S0=110; K=100;

N = 1000;
Y = randn(1,N);          % Normal random variable
S = S0*exp((r-sig^2/2)*T + sig*sqrt(T)*Y);
F = exp(-r*T)*max(0,S-K);
C = exp(-r*T)*S;
Fave = sum(F)/N;
Cave = sum(C)/N;
lam = sum((F-Fave).*(C-Cave)) / sum((C-Cave).^2);
```

Application

MATLAB code, part 2 – control variate estimation:

```
N  = 1e5;
Y  = randn(1,N);          % Normal random variable
S  = S0*exp((r-sig^2/2)*T + sig*sqrt(T)*Y);
F  = exp(-r*T)*max(0,S-K);
C  = exp(-r*T)*S;
F2 = F - lam*(C-S0);

Fave  = sum(F)/N;
F2ave = sum(F2)/N;
sd    = sqrt( sum((F -Fave ).^2)/(N*(N-1)) );
sd2   = sqrt( sum((F2-F2ave).^2)/(N*(N-1)) );
```


Application

Results:

```
>> cv

estimated price (without CV)  = 17.624089 +/- 0.178
estimated price (with CV)     = 17.651112 +/- 0.045
exact price                   = 17.662954
```


Final words

- Monte Carlo estimation is very simple, and efficient when there are multiple underlying assets
- Need to generate correlated Normal random variables
- Central Limit Theorem (CLT) is very important in giving a confidence interval for the computed value
- The use of a control variate with known expected value can greatly reduce the number of MC samples required

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

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

For hedging and risk analysis, we also want to know a whole range of "Greeks" corresponding to first and second derivatives of V with respect to various parameters:

$$\Delta = \frac{\partial V}{\partial S_0}, \qquad \Gamma = \frac{\partial^2 V}{\partial S_0^2},$$
$$\rho = \frac{\partial V}{\partial r}, \qquad \text{Vega} = \frac{\partial V}{\partial \sigma}.$$

Finite difference sensitivities

If $V(\theta) = \mathbb{E}[f(S(T))]$ for an input parameter θ is sufficiently differentiable, then the sensitivity $\dfrac{\partial V}{\partial \theta}$ can be approximated by one-sided finite difference

$$\frac{\partial V}{\partial \theta} = \frac{V(\theta{+}\Delta\theta) - V(\theta)}{\Delta\theta} \; + \; O(\Delta\theta)$$

or by central finite difference

$$\frac{\partial V}{\partial \theta} = \frac{V(\theta{+}\Delta\theta) - V(\theta{-}\Delta\theta)}{2\Delta\theta} \; + \; O((\Delta\theta)^2)$$

(This approach is referred to as getting Greeks by "bumping" the input parameters.)

Finite difference sensitivities

The clear advantage of this approach is that it is very simple to implement (hence the most popular in practice?)

However, the disadvantages are:

- expensive (2 extra sets of calculations for central differences)
- significant bias error if $\Delta\theta$ too large
- machine roundoff errors if $\Delta\theta$ too small
- large variance if $f(S(T))$ discontinuous and $\Delta\theta$ small

Finite difference sensitivities

Let $X^{(i)}(\theta{+}\Delta\theta)$ and $X^{(i)}(\theta{-}\Delta\theta)$ be the values of $f(S(T))$ obtained for different MC samples, so the central difference estimate for $\dfrac{\partial V}{\partial \theta}$ is given by

$$\begin{aligned}\widehat{Y} &= \frac{1}{2\Delta\theta}\left(N^{-1}\sum_{i=1}^{N} X^{(i)}(\theta{+}\Delta\theta) - N^{-1}\sum_{i=1}^{N} X^{(i)}(\theta{-}\Delta\theta)\right) \\ &= \frac{1}{2N\Delta\theta}\sum_{i=1}^{N}\left(X^{(i)}(\theta{+}\Delta\theta) - X^{(i)}(\theta{-}\Delta\theta)\right)\end{aligned}$$

Finite difference sensitivities

If independent samples are taken for both $X^{(i)}(\theta{+}\Delta\theta)$ and $X^{(i)}(\theta{-}\Delta\theta)$ then

$$\begin{aligned}\mathbb{V}[\widehat{Y}] &\approx \left(\frac{1}{2N\Delta\theta}\right)^2 \sum_j \Big(\mathbb{V}[X(\theta{+}\Delta\theta)] + \mathbb{V}[X(\theta{-}\Delta\theta)]\Big) \\ &\approx \left(\frac{1}{2N\Delta\theta}\right)^2 2\,N\,\mathbb{V}[f] \\ &= \frac{\mathbb{V}[f]}{2N(\Delta\theta)^2}\end{aligned}$$

which is very large for $\Delta\theta \ll 1$.

Finite difference sensitivities

It is much better for $X^{(i)}(\theta{+}\Delta\theta)$ and $X^{(i)}(\theta{-}\Delta\theta)$ to use the same set of random inputs.

If $X^{(i)}(\theta)$ is differentiable with respect to θ, then

$$X^{(i)}(\theta{+}\Delta\theta) - X^{(i)}(\theta{-}\Delta\theta) \approx 2\,\Delta\theta\,\frac{\partial X^{(i)}}{\partial\theta}$$

and hence

$$\mathbb{V}[\widehat{Y}] \approx N^{-1}\,\mathbb{V}\left[\frac{\partial X}{\partial \theta}\right],$$

which behaves well for $\Delta\theta \ll 1$, so one should choose a small value for $\Delta\theta$ to minimise the bias due to the finite differencing.

Finite difference sensitivities

However, there are problems if $\Delta\theta$ is chosen to be extremely small.

In finite precision arithmetic, computing $X(\theta)$ has an error which is approximately random with r.m.s. magnitude δ

- single precision $\delta \approx 10^{-6}|X|$
- double precision $\delta \approx 10^{-14}|X|$

Hence, should choose bump $\Delta\theta$ so that

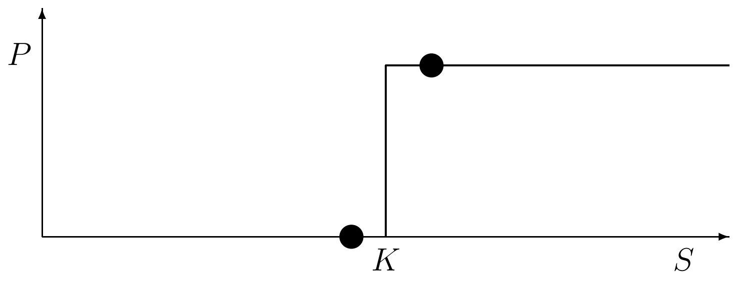
$$|X(\theta{+}\Delta\theta) - X(\theta{-}\Delta\theta)| \gg \delta$$

Most banks probably use double precision to be safe.

Finite difference sensitivities

Next, we analyse the variance of the finite difference estimator when the payoff is discontinuous.

The problem is that a small bump in the asset S can produce a big bump in the payoff – not differentiable



Finite difference sensitivities

What is the probability that $S(\theta \pm \Delta\theta)$ will be on different sides of the discontinuity?

Separation of $S(\theta \pm \Delta\theta)$ is $O(\Delta\theta)$

$\mathbb{P}(\,|S(\theta) - K| < c\,\Delta\theta) = O(\Delta\theta)$

Hence, $O(\Delta\theta)$ probability of straddling the strike.

This leads to a variance which is $O(N^{-1}\Delta\theta^{-1})$.

Finite difference sensitivities

So, small $\Delta\theta$ gives a large variance, while a large $\Delta\theta$ gives a large finite difference discretisation error.

To determine the optimum choice we use this result:

$$\mathbb{E}\left[\left(\widehat{Y} - \mathbb{E}[Y]\right)^2\right] \;=\; \mathbb{V}[\widehat{Y}] + \left(\mathbb{E}[\widehat{Y}] - \mathbb{E}[Y]\right)^2$$

$$\text{Mean Square Error} \;=\; \text{variance} + (\text{bias})^2$$

In our case, we have

$$\mathbb{V}[\widehat{Y}] + (\text{bias})^2 \sim \frac{a}{N\,\Delta\theta} + b\,\Delta\theta^4.$$

which can be minimised by choosing $\Delta\theta$ appropriately.

Pathwise sensitivities

We start with

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

where $p_W(W)$ is the probability density function for $W(T)$, 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.

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

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

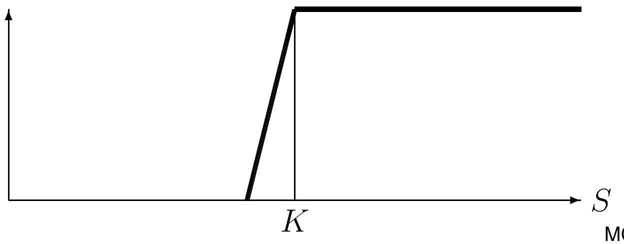
Gives incorrect estimates when $f(S)$ is discontinuous.
e.g. for digital put $\dfrac{\partial f}{\partial S}=0$ so estimated value of Greek is zero – clearly wrong.

Pathwise sensitivities

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)



SDE Path Simulation

So far, we have considered European options with assets satisfying Geometric Brownian Motion SDEs

Now we consider the more general case in which the solution to the SDE needs to be approximated because

- the option is path-dependent, and/or
- the SDE is not integrable

Euler-Maruyama method

The simplest approximation for the scalar SDE

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

is the forward Euler scheme, which is known as the Euler-Maruyama approximation when applied to SDEs:

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

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.

For ODEs, the $O(h)$ accuracy of forward Euler method is considered poor, but for SDEs it is very hard to do better.

Weak convergence

In finance applications, mostly concerned with **weak** errors, the error in the expected payoff. For a European payoff $f(S(T))$ this is

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

and it is of order α if

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

For a path-dependent option, the weak error 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.

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 if $a(S,t)$ and $b(S,t)$ are Lipschitz w.r.t. S, t

$$\|p(S) - \widehat{p}(S)\|_1 = O(h)$$

and hence for bounded payoffs

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

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

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

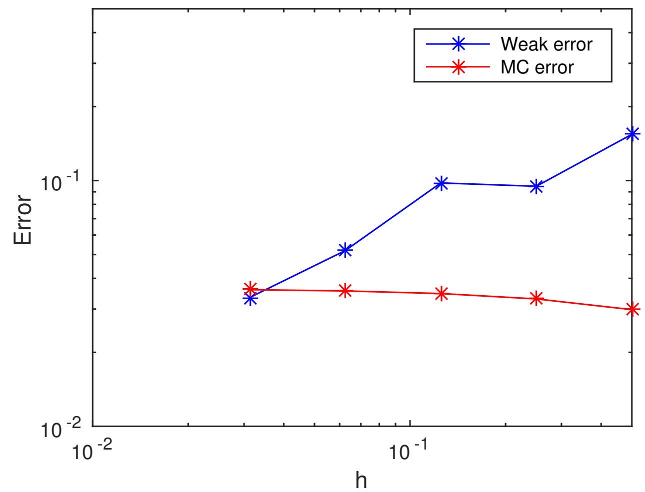
$r = 0.05,\ \sigma = 0.5,\ T = 1$

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

Comparison to exact solution:



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

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

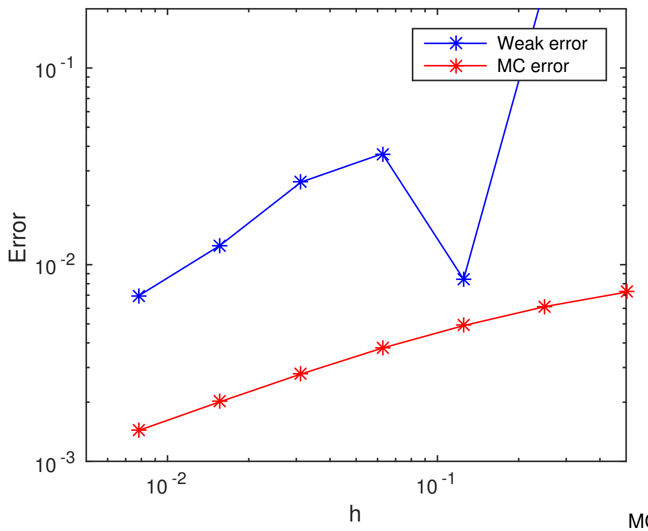
Weak convergence

To minimise the number of paths that need to be simulated, best to 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.

This is like using the same driving Brownian paths for finite difference Greeks. The variance is lower because the h and $2h$ paths are close to each other (**strong** convergence).

Weak convergence

Comparison to $2h$ approximation:



Barrier option

Some path-dependent options give only $O(\sqrt{h})$ weak convergence if the numerical payoff is not constructed carefully.

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

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

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

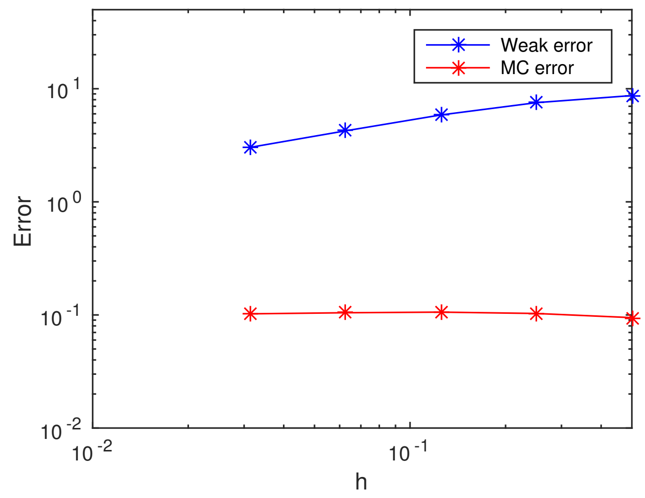
Down-and-out call: $S_0 = 100, K = 110, B = 90.$

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.)

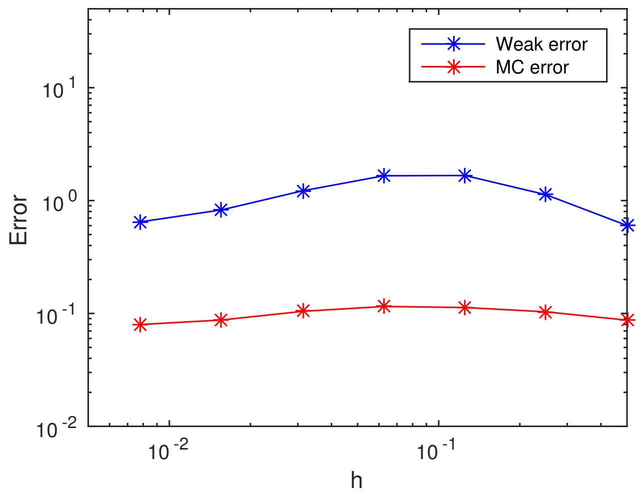
Barrier option

Comparison to exact solution:



Barrier option

Comparison to $2h$ approximation:



Lookback option

A floating-strike lookback call option has discounted payoff

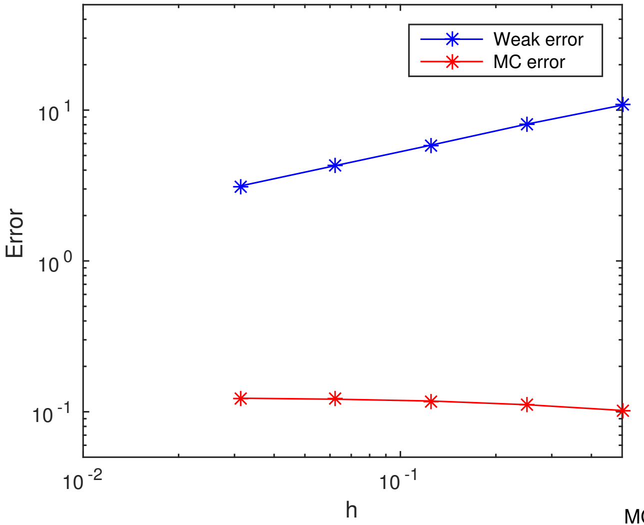
$$\exp(-rT)\ \left(S_T - \min_{[0,T]} S_t \right)$$

The natural numerical discretisation of this is

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

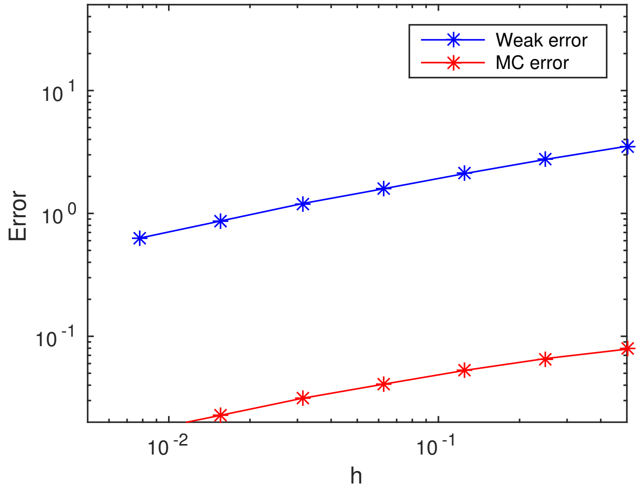
Lookback option

Comparison to exact solution



Lookback option

Comparison to $2h$ approximation



Brownian Bridge

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

Consider simple Brownian motion

$$\mathrm{d}S = a\ \mathrm{d}t + b\ \mathrm{d}W$$

with constant $a,\ b$.

If we know the values S_n, S_{n+1} at times $t_n, t_{n+1} = t_n + h$, the Brownian Bridge construction considers the conditional behaviour of $S(t)$ within the timestep $t_n < t < t_{n+1}$.

Brownian Bridge

Two key results:

- The probability of the minimum dropping below B is:

$$\mathbb{P}(\min_{t_n<t<t_{n+1}} S(t) < B \mid S_n, S_{n+1} > B) = \exp\left(-\frac{2\,(S_{n+1}-B)\,(S_n-B)}{b^2 h}\right)$$

- A sample of the conditional minimum is given by

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

where U_n is 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

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

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

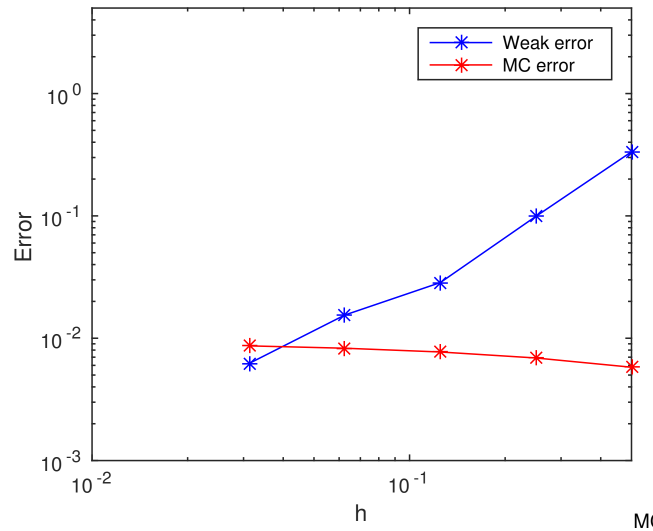
The payoff is then

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

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

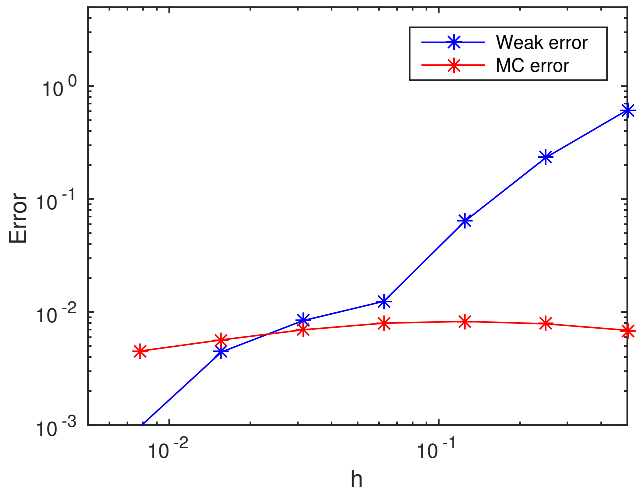
Weak convergence

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.

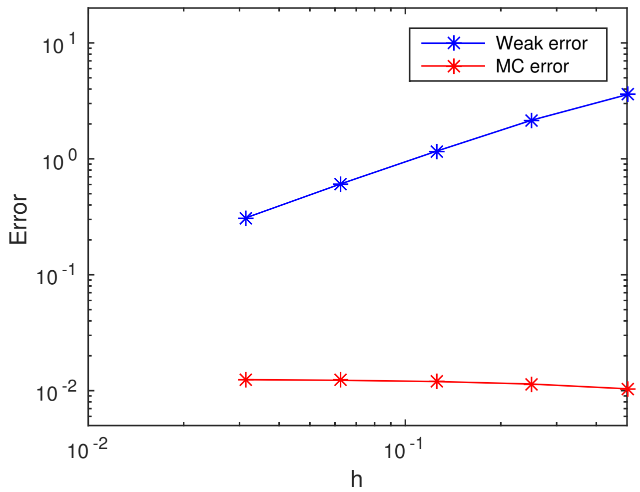
We construct a minimum $\widehat{M}_n$ within each timestep and then the payoff is

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

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

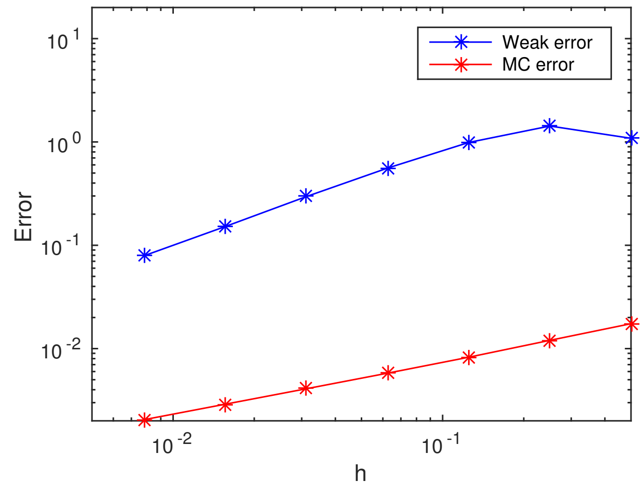
Weak convergence

Lookback: comparison to true solution



Weak convergence

Lookback: h versus $2h$ solution



Final Words

- The Greeks are very important
- “Bumping” is the standard approach, but Pathwise Sensitivity analysis is more efficient and more accurate, provided the payoff does not have a discontinuity

- The Euler-Maruyama method is used for path-dependent payoffs and general SDEs
- It gives $O(h)$ weak convergence – the error (bias) in the expected value
- Care must be taken with some path-dependent options to achieve this order of convergence.