Advanced Monte Carlo Methods: I

Prof. Mike Giles

mike.giles@maths.ox.ac.uk

Oxford University Mathematical Institute

Lecture I

- Improved numerical methods:
  - Euler and Milstein method
  - approximating exotic options
    - Brownian motion interpolation
    - lookback option
    - barrier option
    - Asian option
    - digital option
- Multi-level Monte Carlo method (current research)

Euler scheme

Given the generic SDE:

$$\mathrm{d}S(t) = a(S)\,\mathrm{d}t + b(S)\,\mathrm{d}W(t), \quad 0<t<T,$$

the Euler discretisation with timestep h is:

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

where ΔW_n are Normal with mean 0, variance h.

- How good is this approximation?
- How do the errors behave as $h \to 0$?

These are much harder questions when working with SDEs instead of ODEs.

Weak convergence

For most finance applications, what matters is the **weak** order of convergence, defined by the error in the expected value of the payoff.

For a European option, the weak order is m if

$$\mathbb{E}\left[f(S(T))\right] - \mathbb{E}\left[f(\widehat{S}_N)\right] = O(h^m)$$

The Euler scheme has order 1 weak convergence, so the discretisation "bias" is asymptotically proportional to h.

Strong convergence

In some Monte Carlo applications, what matters is the **strong** order of convergence, defined by the average error in approximating each individual path.

For the generic SDE, the strong order is m if

$$\mathbb{E}\left[\,\left|S(T) - \widehat{S}_N\right|\,\right] = O(h^m)$$

The Euler scheme has order 1/2 strong convergence. The leading order errors are as likely to be positive as negative, and so cancel out – this is why the weak order is higher.

Milstein Scheme

For a scalar problem, the Milstein scheme is

$$\widehat{S}_{n+1} = \widehat{S}_n + a(\widehat{S}_n)\, h + b(\widehat{S}_n)\, \Delta W_n + \tfrac{1}{2}\, b'(\widehat{S}_n)\, b(\widehat{S}_n)(\Delta W_n^2 - h).$$

This comes from performing a stochastic equivalent of a Taylor series expansion.

This gives the same order 1 weak convergence as the Euler scheme, but an improved order 1 strong convergence.

Milstein Scheme

For a multi-dimensional problem, the Milstein scheme is

$$\begin{aligned}\widehat{S}_{i,n+1} &= \widehat{S}_{i,n} + a_i\, h + \sum_j b_{ij}\, \Delta W_{j,n} \\ &+ \sum_{j,k,l} \tfrac{1}{2}\, \frac{\partial b_{ij}}{\partial S_l}\, b_{lk} \left(\Delta W_{j,n}\, \Delta W_{k,n} - h\, \delta_{jk} - A_{jk,n}\right)\end{aligned}$$

where δ_{jk} is the Kronecker delta which equals 1 if $j=k$, and zero otherwise, and $A_{jk,n}$ is the Lévy area defined by

$$A_{jk,n} = \int_{t_n}^{t_{n+1}} (W_j(t) - W_j(t_n))\, \mathrm{d}W_k(t) - (W_k(t) - W_k(t_n))\, \mathrm{d}W_j(t).$$

Milstein Scheme

Simulating the Lévy areas is computationally demanding, which limits the use of the Milstein method.

However, using the anti-symmetry $A_{jk} = -A_{kj}$, the term involving the Lévy areas can be expressed as

$$\sum_{j,k,l} \tfrac{1}{4} \left(\frac{\partial b_{ij}}{\partial S_l}\, b_{lk} - \frac{\partial b_{ik}}{\partial S_l}\, b_{lj} \right) A_{jk,n}$$

which is zero if the SDE satisfies the *commutativity* condition

$$\sum_l \frac{b_{ij}}{\partial S_l}\, b_{lk} = \sum_l \frac{b_{ik}}{\partial S_l}\, b_{lj}.$$

Exotic options

Evaluating European options is straightforward: simulate N paths and average the payoffs based on the terminal state.

There are lots of other options which are harder to approximate – in some cases, care is needed to even get order 1 weak convergence.

To understand the difficulties and develop improved numerical treatment we look at Brownian interpolation.

Brownian interpolation

Simple Brownian motion has constant drift and volatility:

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

If we know the values at two times t_n and t_{n+1}, then at intermediate times

$$t = t_n + \lambda(t_{n+1} - t_n), \quad 0 < \lambda < 1$$

we can combine the equations for $S(t), S(t_n), S(t_{n+1})$ to give

$$\begin{aligned} S(t) &= S(t_n) + \lambda\,(S(t_{n+1}) - S(t_n)) \\ &\quad + b\,(W(t) - W(t_n) - \lambda(W(t_{n+1}) - W(t_n)))\,. \end{aligned}$$

Note: $S(t)$ deviates from a straight-line interpolation if, and only if, $W(t)$ deviates from a straight-line interpolation.

Brownian interpolation

Further analysis leads to the following results:

- $\min_t S(t) = \frac{1}{2}\left(S(t_n) + S(t_{n+1}) - \sqrt{(\Delta S)^2 - 2\,h\,b^2 \log U_n}\right)$

  where U_n is uniformly distributed on $[0,1]$
- Prob$(\min_t S(t) < B) =$
  $\exp\left(\dfrac{-2\,(S(t_n) - B)^+ (S(t_{n+1}) - B)^+}{b^2\,h} \right)$
- $\displaystyle\int_{t_n}^{t_{n+1}} S(t)\ \mathrm{d}t = \tfrac{1}{2}h(S(t_n) + S(t_{n+1})) + b\Delta I$

  where ΔI is Normally distributed, independent of ΔW, with zero mean and variance $h^3/12$

Brownian interpolation

How are these results used?

- calculate the $\widehat{S}_n$ as usual, using the Euler or Milstein schemes.
- use the Brownian motion results to obtain estimates for payoffs which depend on continuous monitoring of the path $S(t)$
- in general, gives better results than linear interpolation of $\widehat{S}$, because $S(t)$ deviates by $O(h^{1/2})$ from straight-path interpolation.

Exotic options

Lookback option:

$$P = \left(S(T) - \min_{0<t<T} S(t) \right).$$

Simple approximation ($O(h^{1/2})$ weak convergence):

$$\widehat{S}_{min} = \min_n \widehat{S}_n$$

Better approximation ($O(h)$ weak convergence):

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

Exotic options

Barrier option – down-and-out call:

$$P = \mathbf{1}(\min_{0<t<T} S(t) > B)\ \max(0, S(T) - K)$$

Simple approximation ($O(h^{1/2})$ weak convergence):

$$P = \mathbf{1}(\min_n \widehat{S}_n > B)\ \max(0, \widehat{S}_N - K)$$

Better approximation ($O(h)$ weak convergence):

$$P = \prod_n \left(1 - \exp\left(\frac{-2\, (\widehat{S}_n - B)^+ (\widehat{S}_{n+1} - B)^+}{b_n^2\, h} \right) \right) \max(0, \widehat{S}_N - K)$$

Exotic options

Asian option: $P = \max\left(0,\ T^{-1} \int_0^T S(t)\ \mathrm{d}t - K \right)$

Simple approximation ($O(h)$ weak convergence):

$$\overline{\widehat{S}} = T^{-1} \sum_1^N \tfrac{1}{2} h\, (\widehat{S}_n + \widehat{S}_{n-1})$$

Better approximation ($O(h)$ weak convergence):

$$\overline{\widehat{S}} = T^{-1} \sum_1^N \tfrac{1}{2} h\, (\widehat{S}_n + \widehat{S}_{n-1}) + b_n \Delta I_n$$

with each ΔI_n a $N(0, h^3/12)$ Normal variable

Exotic options

Digital call option:

$$P = \mathbf{1}(S(T) - K).$$

Simple approximation ($O(h)$ weak convergence):

$$\widehat{P} = \mathbf{1}(\widehat{S}_N - K)$$

Better approximation ($O(h)$ weak convergence and differentiable) based on Brownian motion approximation for final timestep:

$$\widehat{P} = \Phi\left(\frac{\widehat{S}_{N-1} + a_{N-1} h - K}{b_{N-1} \sqrt{h}} \right)$$

where $\Phi(z)$ is the cumulative Normal distribution

Exotic options

Final words:

- simplest approximation of the payoff function may not be best
- improved approximations often possible based on analytic results for simple Brownian motion
- in real applications, options may not be based on continuous monitoring, so may have to use additional corrections

Multilevel Monte Carlo

When solving PDEs, multigrid combines calculations on a nested sequence of grids to get the accuracy of the finest grid at a much lower computational cost.

We will use a similar idea to achieve variance reduction in Monte Carlo path calculations, combining simulations with different numbers of timesteps – same accuracy as finest calculations, but at a much lower computational cost.

Generic Problem

SDE with general drift and volatility terms:

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

Suppose we want to compute the expected value of a European option

$$P = f(S(T))$$

with a uniform Lipschitz bound,

$$|f(U) - f(V)| \ \le c\,\|U-V\|\,, \quad \forall\, U, V.$$

Standard MC Approach

Euler discretisation with timestep h:

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

where ΔW_n are Normal with mean 0, variance h.

Simplest estimator for expected payoff is an average of N independent path simulations:

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

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

Standard MC Approach

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

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

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

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

Aim is to improve this cost to $O\left(\varepsilon^{-2}(\log\varepsilon)^2\right)$

Multilevel MC Approach

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

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

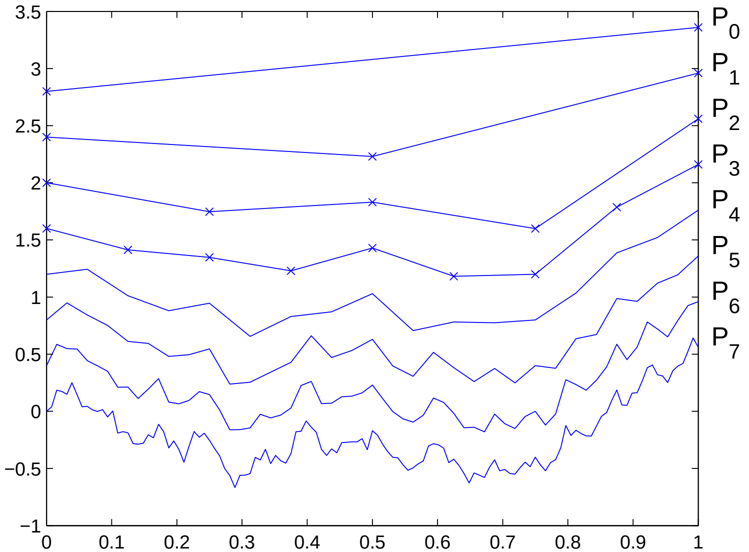
Expected value is same – aim is to reduce variance of estimator for a fixed computational cost.

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

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

Multilevel MC Approach

Discrete Brownian path at different levels



Multilevel MC Approach

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

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

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

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

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

Multilevel MC Approach

For the Euler discretisation and the Lipschitz payoff function

$$\mathbb{V}[\widehat{P}_l - P] = O(h_l) \quad \Longrightarrow \quad \mathbb{V}[\widehat{P}_l - \widehat{P}_{l-1}] = O(h_l)$$

and the optimal N_l is asymptotically proportional to h_l.

To make the combined variance $O(\varepsilon^2)$ requires

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

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

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

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

Multilevel MC Approach

Theorem: *Let P be a functional of the solution of a stochastic o.d.e., and $\widehat{P}_l$ the discrete approximation using a timestep $h_l = M^{-l}\, T$.*

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

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

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

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

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

$$C_l \leq c_3\, N_l\, h_l^{-1}$$

Multilevel MC Approach

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

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

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

with a computational complexity C with bound

$$C \leq \begin{cases} c_4\, \varepsilon^{-2}, & \beta > 1, \\ c_4\, \varepsilon^{-2} (\log \varepsilon)^2, & \beta = 1, \\ c_4\, \varepsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1. \end{cases}$$

Convergence Test

Asymptotically,

$$\mathbb{E}[\widehat{P}_L - \widehat{P}_{L-1}] \approx (M-1)\, \mathbb{E}[P - \widehat{P}_L]$$

so this can be used to decide when the bias error is sufficiently small.

In case the correction changes sign at some level, it is safer to use the convergence test

$$\max\left\{ M^{-1} \left|\widehat{Y}_{L-1}\right|, \left|\widehat{Y}_L\right| \right\} < (M-1)\, \frac{\varepsilon}{\sqrt{2}}.$$

Multilevel Algorithm

1. start with $L=0$
2. estimate V_L using an initial $N_L=10^4$ samples
3. define optimal $N_l,\ l=0,\ldots,L$
4. evaluate extra samples as needed for new N_l
5. if $L\geq 2$, test for convergence
6. if $L<2$ or not converged, set $L:=L+1$ and go to 2.

Numerical results use $M=4$, which is almost twice as efficient as $M=2$.

Results

Geometric Brownian motion:

$$\mathrm{d}S = r\,S\,\mathrm{d}t + \sigma\,S\,\mathrm{d}W, \qquad 0<t<1,$$

$S(0)=1$, $r=0.05$, $\sigma=0.2$

Heston model:

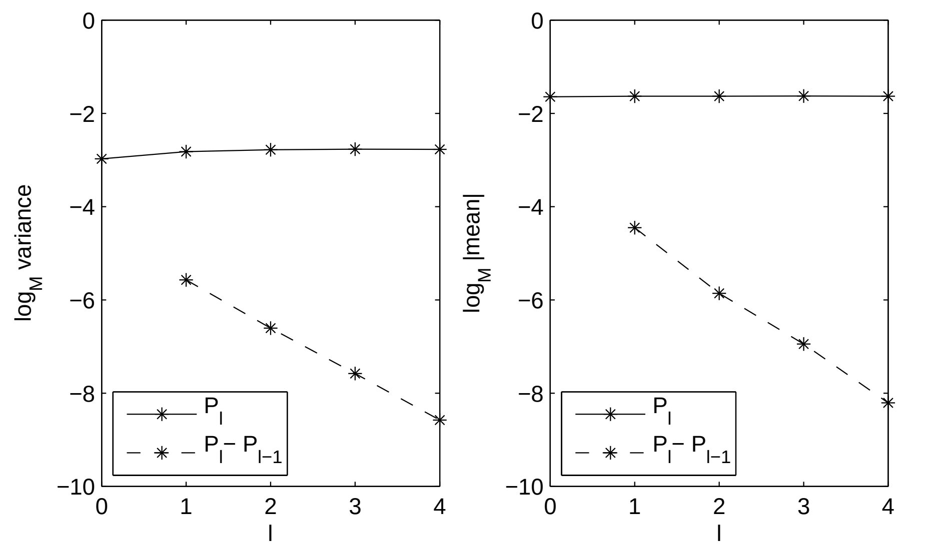
$$\begin{aligned} \mathrm{d}S &= r\,S\,\mathrm{d}t + \sqrt{V}\,S\,\mathrm{d}W_1, \qquad 0<t<1 \\ \mathrm{d}V &= \lambda\,(\sigma^2 - V)\,\mathrm{d}t + \xi\,\sqrt{V}\;\mathrm{d}W_2, \end{aligned}$$

$S(0)=1$, $V(0)=0.04$, $r=0.05$, $\sigma=0.2$, $\lambda=5$, $\xi=0.25$, $\rho=-0.5$

All calculations use $M=4$, more efficient than $M=2$.

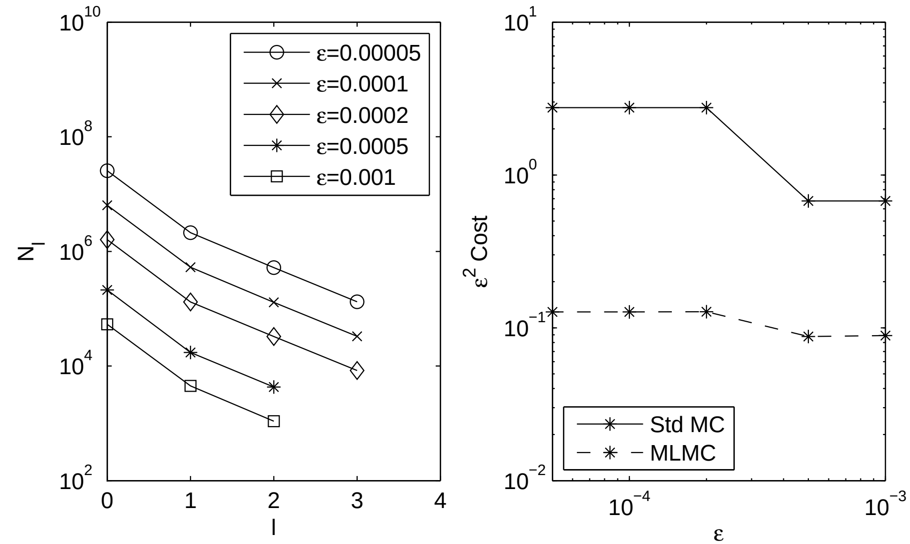
Results

GBM: European call, $\max(S(1)-1,0)$



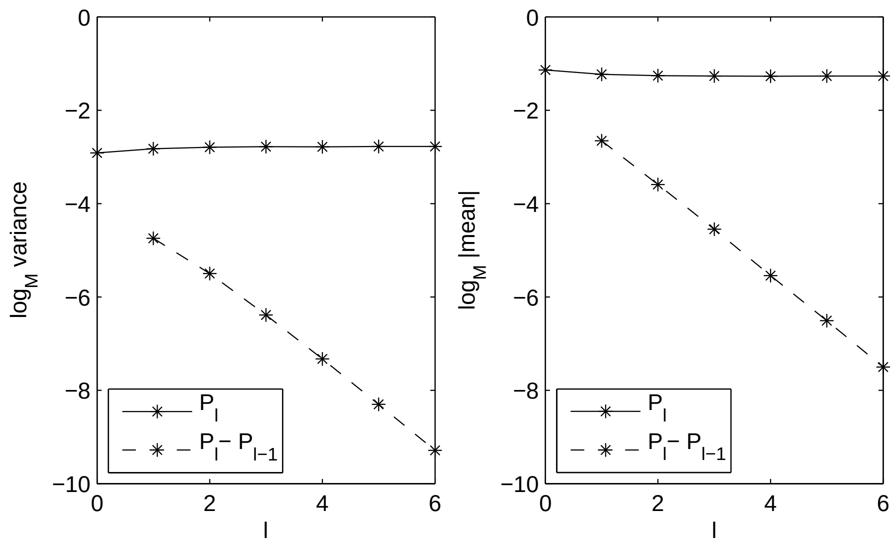
Results

GBM: European call, $\max(S(1)-1,0)$



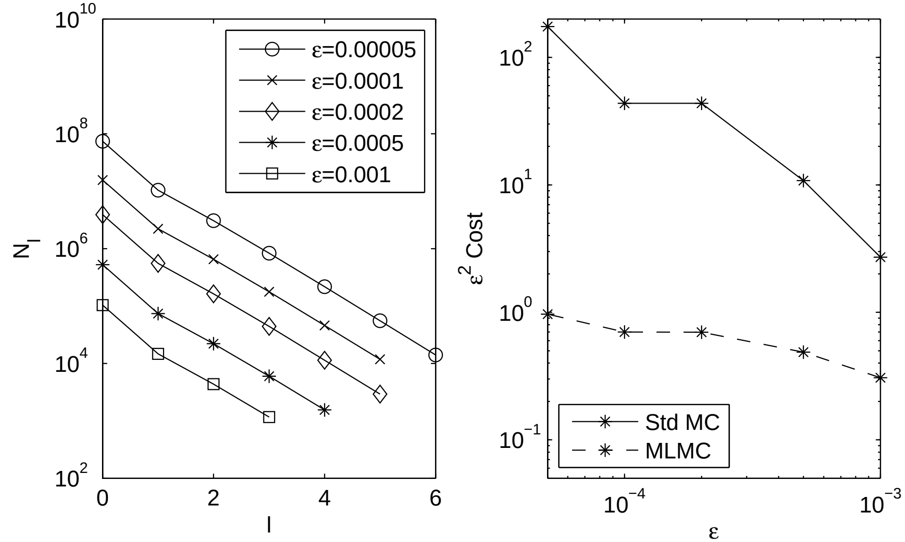
Results

GBM: lookback option, $S(1) - \min_{0<t<1} S(t)$



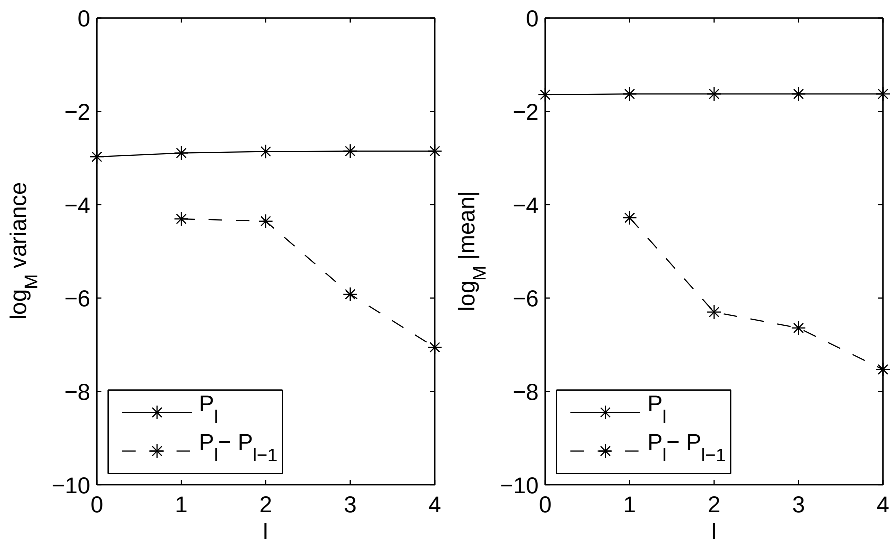
Results

GBM: lookback option, $S(1) - \min_{0<t<1} S(t)$



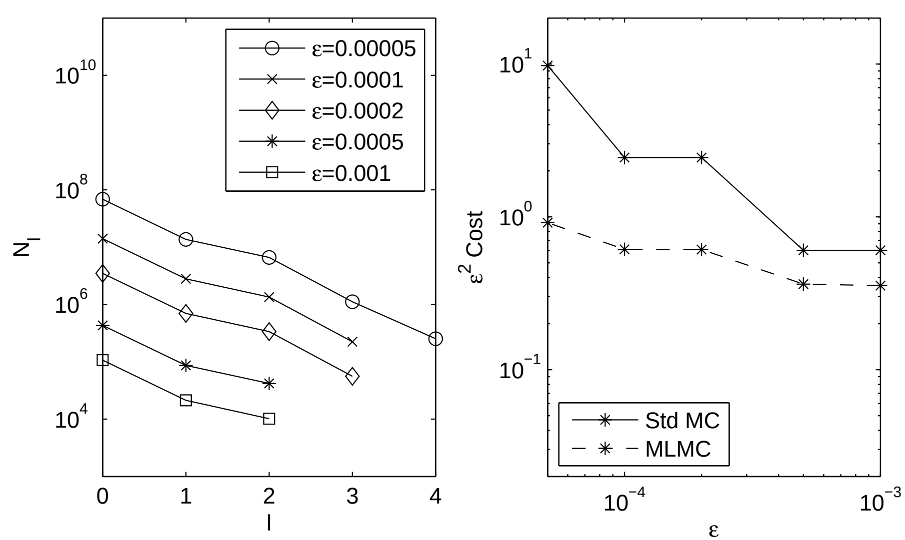
Results

Heston model: European call



Results

Heston model: European call



Milstein Scheme

The theorem suggests use of Milstein approximation
– better strong convergence, same weak convergence

Generic scalar SDE:

$$\mathrm{d}S(t) = a(S,t)\ \mathrm{d}t + b(S,t)\ \mathrm{d}W(t), \quad 0<t<T.$$

Milstein scheme:

$$\widehat{S}_{n+1} = \widehat{S}_n + a\,h + b\,\Delta W_n + \tfrac{1}{2}\,b'\,b\left((\Delta W_n)^2 - h\right).$$

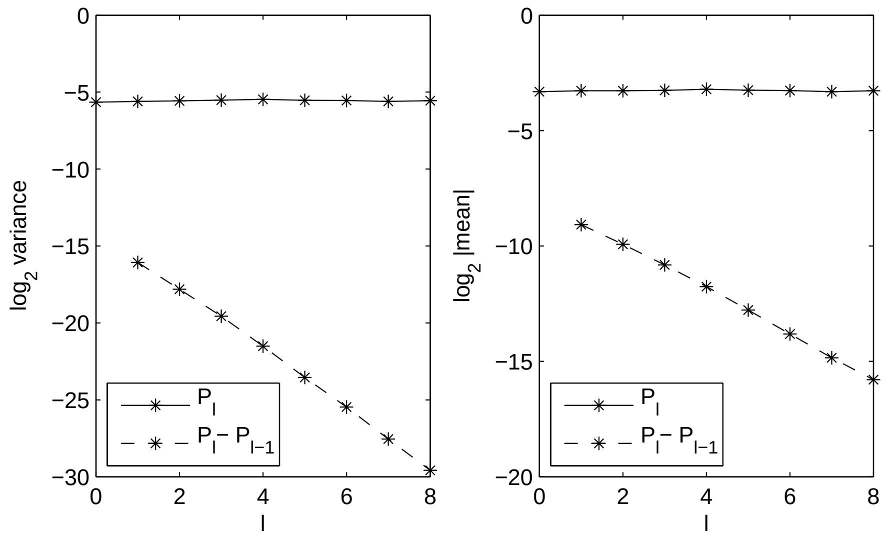
Milstein Scheme

In scalar case:

- $O(h)$ strong convergence
- $O(\varepsilon^{-2})$ complexity for Lipschitz payoffs – trivial
- $O(\varepsilon^{-2})$ complexity for more complex cases using carefully constructed estimators based on Brownian interpolation or extrapolation
  - digital, with discontinuous payoff
  - Asian, based on average
  - lookback and barrier, based on min/max

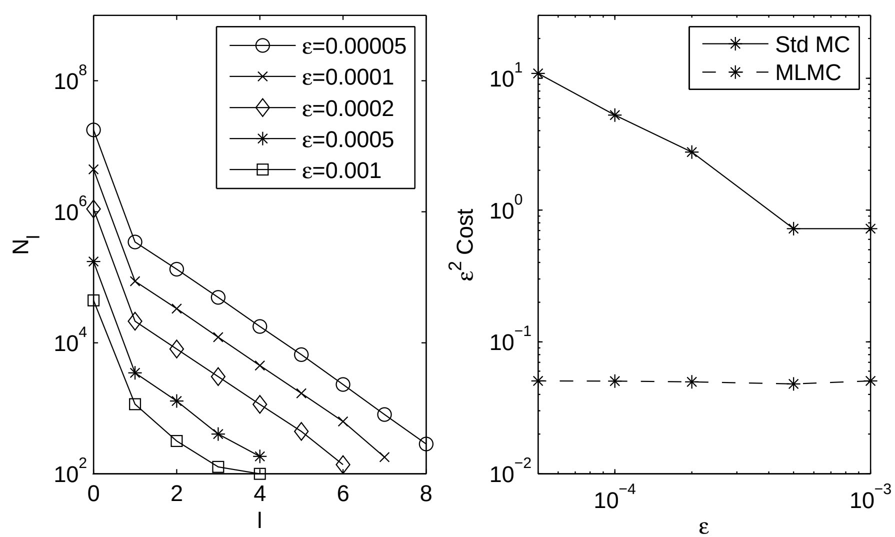
MLMC Results

GBM: European call, $\exp(-rT)\ \max(S(T)-K,0)$



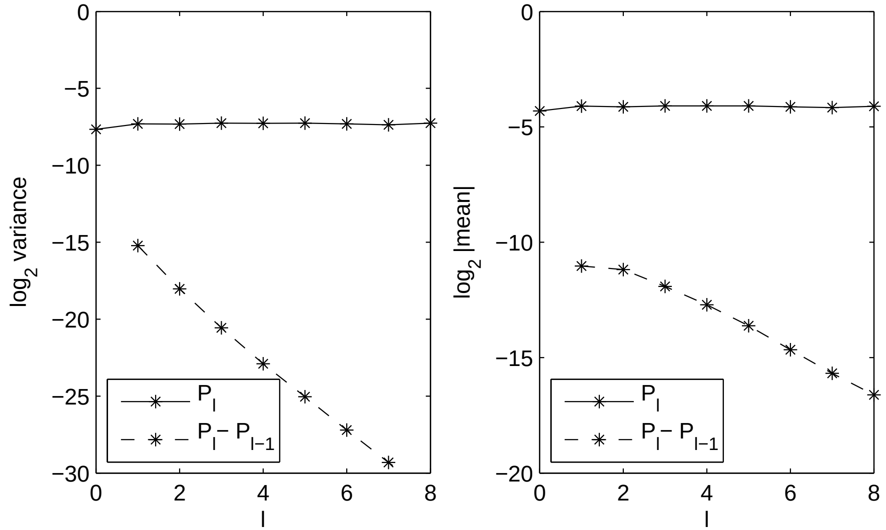
MLMC Results

GBM: European call, $\exp(-rT)\ \max(S(T)-K,0)$



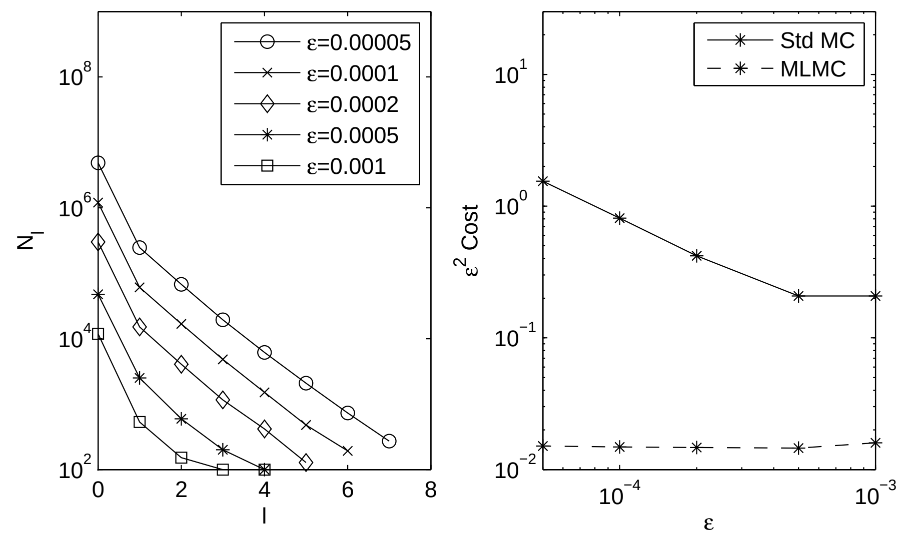
MLMC Results

GBM: Asian option, $\exp(-rT)\,\max(T^{-1}\int_0^T S(t)\,\mathrm{d}t - 1, 0)$



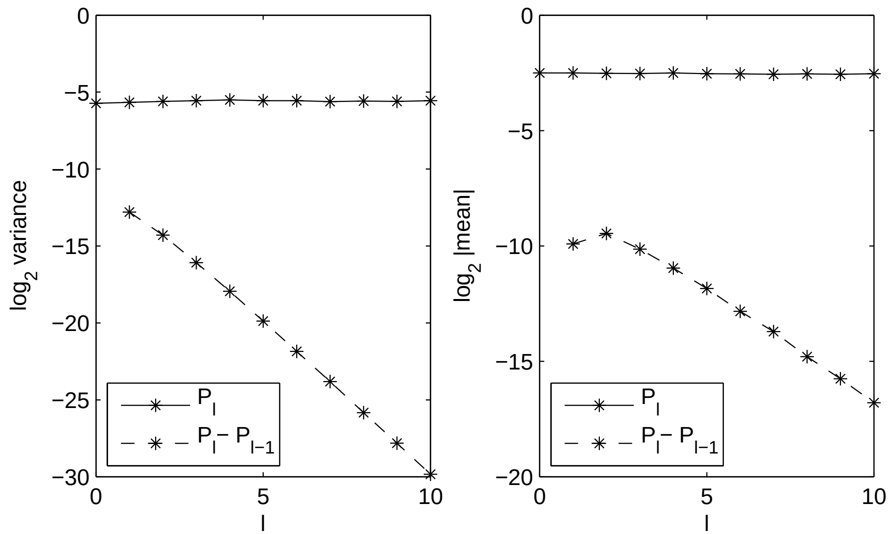
MLMC Results

GBM: Asian option, $\exp(-rT)\,\max(T^{-1}\int_0^T S(t)\,\mathrm{d}t - 1, 0)$



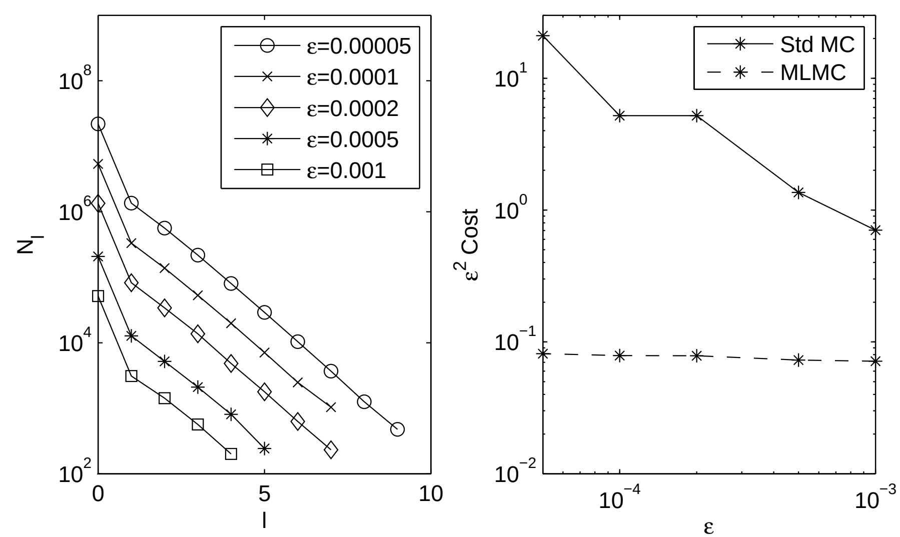
MLMC Results

GBM: lookback option, $\exp(-rT)\,(S(T) - \min_{0<t<T} S(t))$



MLMC Results

GBM: lookback option, $\exp(-rT)\,(S(T) - \min_{0<t<T} S(t))$



Milstein Scheme

Generic vector SDE:

$$\mathrm{d}S(t) = a(S,t)\,\mathrm{d}t + b(S,t)\,\mathrm{d}W(t), \quad 0<t<T,$$

with correlation matrix $\Omega(S,t)$ between elements of $\mathrm{d}W(t)$.

Milstein scheme:

$$\begin{aligned}\widehat{S}_{i,n+1} &= \widehat{S}_{i,n} + a_i\,h + b_{ij}\,\Delta W_{j,n} \\ &\quad + \tfrac{1}{2}\,\frac{\partial b_{ij}}{\partial S_l}\,b_{lk}\left(\Delta W_{j,n}\,\Delta W_{k,n} - h\,\Omega_{jk} - A_{jk,n}\right)\end{aligned}$$

with implied summation, and Lévy areas defined as

$$A_{jk,n} = \int_{t_n}^{t_{n+1}} (W_j(t)-W_j(t_n))\,\mathrm{d}W_k - (W_k(t)-W_k(t_n))\,\mathrm{d}W_j.$$

Milstein Scheme

In vector case:

- $O(h)$ strong convergence if Lévy areas are simulated correctly – expensive
- $O(h^{1/2})$ strong convergence in general if Lévy areas are omitted, except if a certain commutativity condition is satisfied (useful for a number of real cases)
- Lipschitz payoffs can be handled well using antithetic variables
- Other cases may require approximate simulation of Lévy areas

Results

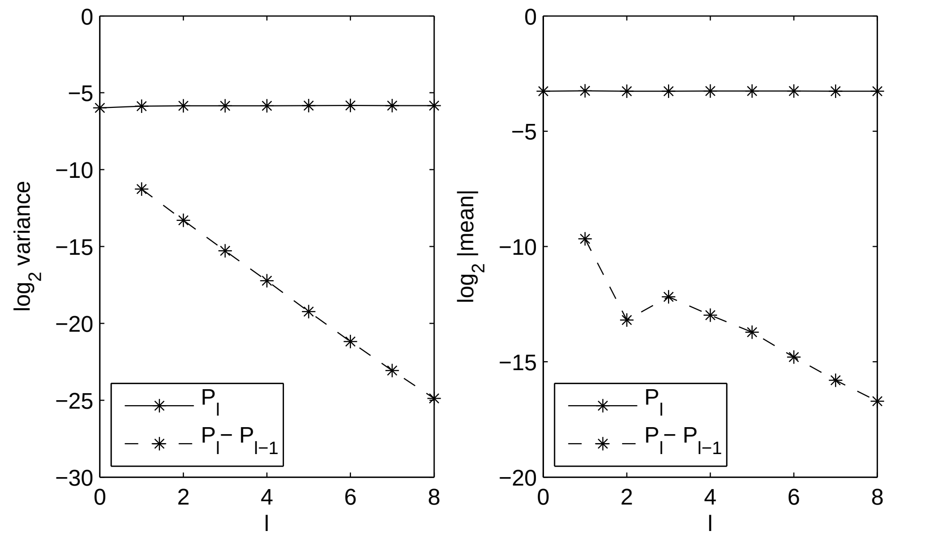
Heston model:

$$\begin{aligned}\mathrm{d}S &= r\,S\,\mathrm{d}t + \sqrt{V}\,S\,\mathrm{d}W_1, \qquad 0<t<T \\ \mathrm{d}V &= \lambda\,(\sigma^2 - V)\,\mathrm{d}t + \xi\,\sqrt{V}\,\mathrm{d}W_2,\end{aligned}$$

$T=1,\;\; S(0)=1,\;\; V(0)=0.04,\;\; r=0.05,$
$\sigma=0.2,\;\; \lambda=5,\;\; \xi=0.25,\;\; \rho=-0.5$

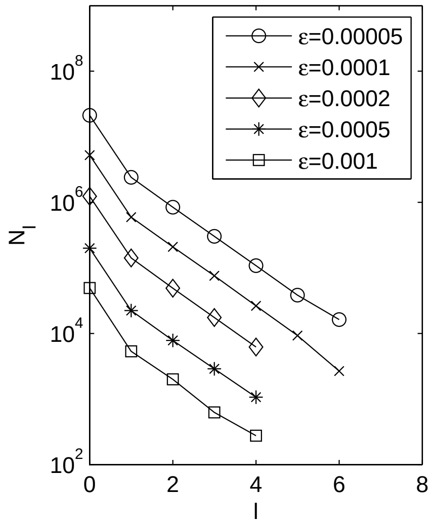
MLMC Results

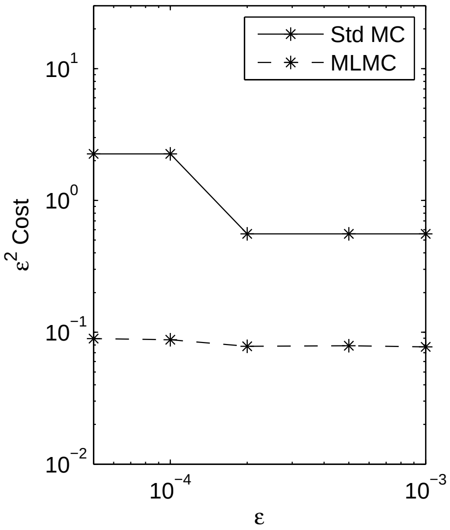
Heston model: European call



MLMC Results

Heston model: European call





Extensions

- Quasi-Monte Carlo – very effective on coarse grids and reduces overall cost to roughly $O(\varepsilon^{-1.5})$ in simplest cases
- multivariate discontinuous payoffs – simplest approach is to use “splitting” for multiple simulations of final timestep
- multivariate SDEs requiring Lévy areas – preliminary results look encouraging
- jump-diffusion models – preliminary results look encouraging
- Greeks – work in progress
- American options – work in progress

Conclusions

Multilevel Monte Carlo method has already achieved

- improved order of complexity
- significant benefits for model problems

but more research is needed, both theoretical and applied.

M.B. Giles, “Multi-level Monte Carlo path simulation”,
Operations Research, 56(3):607-617, 2008.

M.B. Giles. “Improved multilevel Monte Carlo convergence using the Milstein scheme”, pages 343-358 in Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer, 2007.

`people.maths.ox.ac.uk/~gilesm/finance.html`

Email: `mike.giles@maths.ox.ac.uk`

Advanced Monte Carlo Methods: II

Prof. Mike Giles

mike.giles@maths.ox.ac.uk

Oxford University Mathematical Institute

Lecture II

Computing Greeks

- finite differences
- likelihood ratio method
- pathwise sensitivities
  - standard treatment
  - adjoint evaluation (current research)

SDE path simulation

For the generic stochastic differential equation

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

an Euler approximation with timestep h is

$$\widehat{S}_{n+1} = \widehat{S}_n + a(\widehat{S}_n)\,h + b(\widehat{S}_n)\,Z_n\,\sqrt{h},$$

where Z is a $N(0,1)$ random variable. To estimate the value of a European option

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

we take the average of N paths with M timsteps:

$$\widehat{V} = N^{-1}\sum_i f(\widehat{S}_M^{(i)}).$$

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:

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

These are needed for hedging and for risk analysis.

Finite difference sensitivities

If $V(\theta) = \mathbb{E}[f(S(T))]$ for a particular value of an input parameter θ, 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

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

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
- 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,

$$X^{(i)}(\theta\pm\Delta\theta)$$

has an error which is approximately random with r.m.s. magnitude δ

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

Finite difference sensitivities

Consequently,

$$\frac{1}{2\Delta\theta}\left(X^{(i)}(\theta+\Delta\theta) - X^{(i)}(\theta-\Delta\theta)\right)$$

has an extra error term with approximate variance

$$\frac{\delta^2}{2(\Delta\theta)^2}$$

and therefore $\widehat{Y}$ has an extra error term with approximate variance

$$\frac{\delta^2}{2N(\Delta\theta)^2}.$$

Finite difference sensitivities

For double precision computations, if $\theta = O(1)$, then can probably use

$$\Delta\theta = 10^{-6}$$

without any problems, and even the $O(\Delta\theta)$ finite difference error from one-sided differencing will probably be small compared to the MC sampling error.

For single precision, better to use a larger perturbation, e.g.

$$\Delta\theta = 10^{-4}$$

and use the more expensive central differencing to minimise the discretisation error.

Finite difference sensitivities

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

In this case

- For most samples, $X^{(i)}(\theta{+}\Delta\theta) - X^{(i)}(\theta{-}\Delta\theta) = O(\Delta\theta)$
- For an $O(\Delta\theta)$ fraction, $X^{(i)}(\theta{+}\Delta\theta) - X^{(i)}(\theta{-}\Delta\theta) = O(1)$

$$\Longrightarrow \quad \mathbb{E}\left[\frac{X^{(i)}(\theta{+}\Delta\theta) - X^{(i)}(\theta{-}\Delta\theta)}{2\Delta\theta}\right] = O(1)$$

$$\mathbb{E}\left[\left(\frac{X^{(i)}(\theta{+}\Delta\theta) - X^{(i)}(\theta{-}\Delta\theta)}{2\Delta\theta}\right)^2\right] = O(\Delta\theta^{-1})$$

This gives $\mathbb{E}[\widehat{Y}] = O(1)$, but $\mathbb{V}[\widehat{Y}] = O(N^{-1}\Delta\theta^{-1})$.

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 the following result: if $\widehat{Y}$ is an estimator for $\mathbb{E}[Y]$ then

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

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

Finite difference sensitivities

In our case, the MSE (mean-square-error) is

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

This is minimised by choosing $\Delta\theta \propto N^{-1/5}$, giving

$$\sqrt{\text{MSE}} \;\propto\; N^{-2/5}$$

in contrast to the usual MC result in which

$$\sqrt{\text{MSE}} \;\propto\; N^{-1/2}$$

Finite difference sensitivities

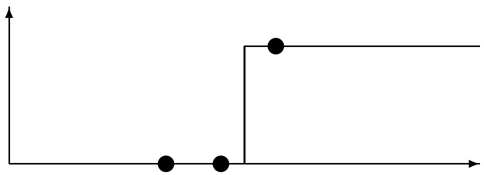
Second derivatives such as Γ can also be approximated by central differences:

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

This will again have a larger variance if either the payoff or its derivative is discontinuous.

Finite difference sensitivities

Discontinuous payoff:



For an $O(\Delta\theta)$ fraction of samples

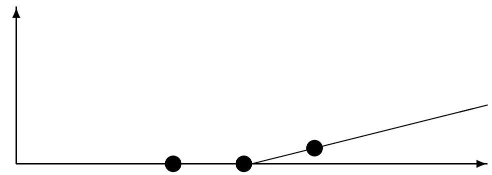
$$X^{(i)}(\theta{+}\Delta\theta) - 2X^{(i)}(\theta) + X^{(i)}(\theta{-}\Delta\theta) = O(1)$$

$$\Longrightarrow \quad \mathbb{E}\left[\left(\frac{X^{(i)}(\theta{+}\Delta\theta) - 2X^{(i)}(\theta) + X^{(i)}(\theta{-}\Delta\theta)}{\Delta\theta^2}\right)^2\right] = O(\Delta\theta^{-3})$$

This gives $\mathbb{V}[\widehat{Y}] = O(N^{-1}\Delta\theta^{-3})$.

Finite difference sensitivities

Discontinuous derivative:



For an $O(\Delta\theta)$ fraction of samples

$$X^{(i)}(\theta{+}\Delta\theta) - 2X^{(i)}(\theta) + X^{(i)}(\theta{-}\Delta\theta) = O(\theta)$$

$$\Longrightarrow \quad \mathbb{E}\left[\left(\frac{X^{(i)}(\theta{+}\Delta\theta) - 2X^{(i)}(\theta) + X^{(i)}(\theta{-}\Delta\theta)}{\Delta\theta^2}\right)^2\right] = O(\Delta\theta^{-1})$$

This gives $\mathbb{V}[\widehat{Y}] = O(N^{-1}\Delta\theta^{-1})$.

Finite difference sensitivities

Hence, for second derivatives the variance of the finite difference estimator is

- $O(N^{-1})$ if the payoff is twice differentiable
- $O(N^{-1}\Delta\theta^{-1})$ if the payoff has a discontinuous derivative
- $O(N^{-1}\Delta\theta^{-3})$ if the payoff is discontinuous

These can be used to determine the optimum $\Delta\theta$ in each case to minimise the Mean Square Error.

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

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

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

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

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

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

$$\begin{aligned}\frac{\partial \log p}{\partial \sigma} &= -\frac{1}{\sigma} - \sqrt{T}\left(\frac{\log(S/S_0)-(r-\frac{1}{2}\sigma^2)T}{S_0\sigma^2 T}\right)\\ &\quad + \frac{1}{\sigma}\left(\frac{\log(S/S_0)-(r-\frac{1}{2}\sigma^2)T}{S_0\sigma^2 T}\right)^2\end{aligned}$$

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 LRM 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

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

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

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

$$p(\widehat{S}) = \prod_n p_n(\widehat{S}_{n+1}|\widehat{S}_n)$$

$$\log p(\widehat{S}) = \sum_n \log p_n(\widehat{S}_{n+1}|\widehat{S}_n)$$

Likelihood Ratio Method

For the Euler approximation of GBM,

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

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

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

Under certain conditions (e.g. $f(S), a(S,t), b(S,t)$ all continuous and piecewise differentiable)

$$\frac{\partial}{\partial \theta}\, \mathbb{E}[f(S(T))] = \mathbb{E}\left[\frac{\partial f(S(T))}{\partial \theta} \right] = \mathbb{E}\left[\frac{\partial f}{\partial S}\, \frac{\partial S(T)}{\partial \theta} \right].$$

with $\dfrac{\partial S(T)}{\partial \theta}$ computed by differentiating the path evolution.

Pros:

- less expensive (1 cheap calculation for each sensitivity)
- no bias

Cons:

- can't handle discontinuous payoffs

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.

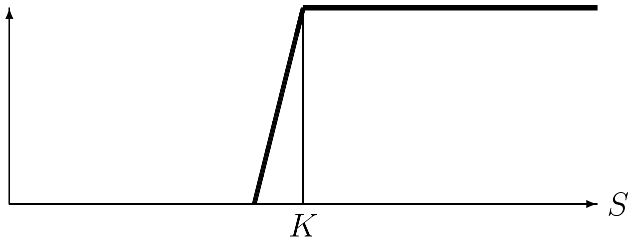
However, this requires $f(S)$ to have a continuous first derivative – a problem in practice

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 sensitivities

Returning to the generic stochastic differential equation

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

an Euler approximation with timestep h gives

$$\widehat{S}_{n+1} = F_n(\widehat{S}_n) \equiv \widehat{S}_n + a(\widehat{S}_n)\,h + b(\widehat{S}_n)\,Z_n\,\sqrt{h}.$$

Defining $\Delta_n = \dfrac{\partial \widehat{S}_n}{\partial S_0}$, then $\Delta_{n+1} = D_n\,\Delta_n$, where

$$D_n \;\equiv\; \frac{\partial F_n}{\partial \widehat{S}_n} \;=\; I + \frac{\partial a}{\partial S}\,h + \frac{\partial b}{\partial S}\,Z_n\,\sqrt{h}.$$

Pathwise sensitivities

The payoff sensitivity to the initial state (Deltas) is then

$$\frac{\partial f(\widehat{S}_N)}{\partial S_0} = \frac{\partial f(\widehat{S}_N)}{\partial \widehat{S}_N}\;\Delta_N$$

If $S(0)$ is a vector of dimension m, then each timestep

$$\Delta_{n+1} = D_n\,\Delta_n,$$

involves a $m\times m$ matrix multiplication, with $O(m^3)$ CPU cost – costly, but still cheaper than finite differences which are also $O(m^3)$ but with a larger coefficient.

Pathwise sensitivities

To calculate the sensitivity to other parameters (such as volatility $\Longrightarrow$ vegas) consider a generic parameter θ.

Defining $\Theta_n = \partial \widehat{S}_n / \partial\theta$, then

$$\Theta_{n+1} = \frac{\partial F_n}{\partial \widehat{S}_n}\, \Theta_n + \frac{\partial F_n}{\partial \theta} \equiv D_n\, \Theta_n + B_n,$$

and hence

$$\frac{\partial f}{\partial \theta} = \frac{\partial f(\widehat{S}_N)}{\partial \widehat{S}_N}\, \Theta_N$$

LIBOR Market Model

As an example, consider the LIBOR market model of BGM, with $m{+}1$ bond maturities T_i, with spacings $T_{i+1} - T_i = \delta_i$.

The forward rate for the interval $[T_i, T_{i+1})$ satisfies

$$\frac{\mathrm{d}L_i(t)}{L_i(t)} = \mu_i(L(t))\,\mathrm{d}t + \sigma_i^\top\,\mathrm{d}W(t), \quad 0 \le t \le T_i,$$

where
$$\mu_i(L(t)) = \sum_{j=\eta(t)}^{i} \frac{\sigma_i^\top \sigma_j\, \delta_j L_j(t)}{1+\delta_j L_j(t)},$$

and $\eta(t)$ is the index of the next maturity date.

For simplicity, we keep $L_i(t)$ constant for $t > T_i$, and take the volatilities to be a function of the time to maturity,

$$\sigma_i(t) = \sigma_{i-\eta(t)+1}(0).$$

LMM implementation

Applying the Euler scheme to the logarithms of the forward rates yields

$$L_{i,n+1} = L_{i,n} \exp\left([\mu_i(L_n) - \|\sigma_i\|^2/2]h + \sigma_i^T Z_n \sqrt{h} \right).$$

For efficiency, we first compute

$$S_{i,n} = \sum_{k=\eta(t)}^{i} \frac{\sigma_k \delta_k L_{k,n}}{1+\delta_k L_{k,n}},$$

and then obtain $\mu_i = \sigma_i^T S_i$.

Each timestep, there is an $O(m)$ cost in computing the S_i's, and then an $O(m)$ cost in updating the L_i's.

LMM implementation

Defining $\Delta_{ij,n} = \partial L_{i,n} / \partial L_{j,0}$, differentiating the Euler scheme yields

$$\Delta_{ij,n+1} = \frac{L_{i,n+1}}{L_{i,n}}\, \Delta_{ij,n} + L_{i,n+1}\, \sigma_i^T S_{ij,n}\, h,$$

where

$$S_{ij,n} = \sum_{k=\eta(nh)}^{i} \frac{\sigma_k\, \delta_k\, \Delta_{kj,n}}{(1+\delta_k L_{k,n})^2}.$$

Each timestep, there is an $O(m^2)$ cost in computing the S_{ij}'s, and then an $O(m^2)$ cost in updating the Δ_{ij}'s.

(Note: programming implementation requires only multiplication and addition – very rapid on modern CPU's).

LIBOR Market Model

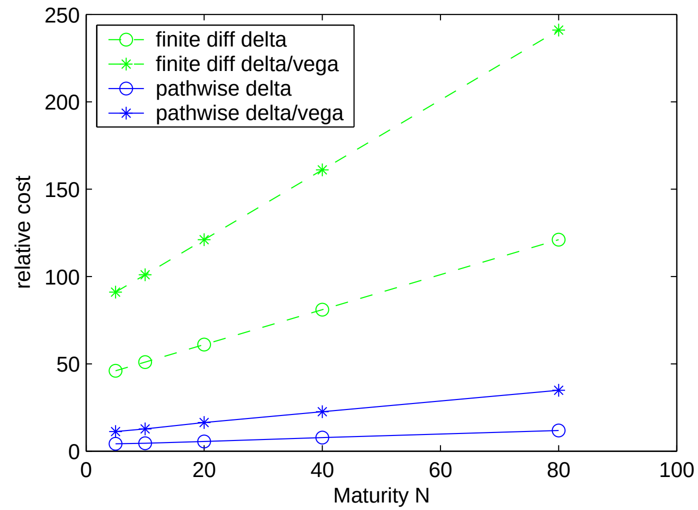
LMM portfolio has 15 swaptions all expiring at the same time, N periods in the future, involving payments/rates over an additional 40 periods in the future.

Interested in computing Deltas, sensitivity to initial $N+40$ forward rates, and Vegas, sensitivity to initial $N+40$ volatilities.

Focus is on the cost of calculating the portfolio value and the sensitivities, relative to just the value.

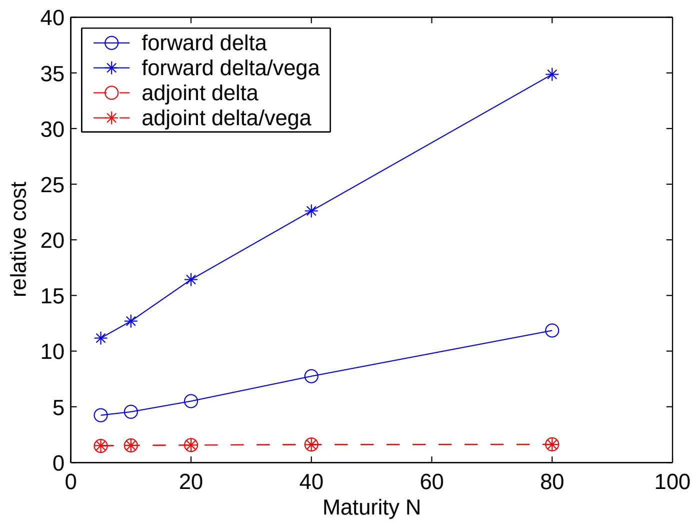
LIBOR Market Model

Finite differences versus forward pathwise sensitivities:



LIBOR Market Model

Forward versus adjoint pathwise sensitivities:



Generic adjoint approach

The adjoint (or dual or reverse mode) approach computes the same values as the forward pathwise approach, but much more efficiently for the sensitivity of a single output to multiple inputs.

The approach has a long history in applied math and engineering:

- optimal control theory (find control which achieves target and minimizes cost);
- design optimization (find shape which maximizes performance);
- primal/dual variables in linear programming optimization.

Generic adjoint approach

Returning to the generic stochastic o.d.e.

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

with Euler approximation

$$\widehat{S}_{n+1} = F_n(\widehat{S}_n) \equiv \widehat{S}_n + a(\widehat{S}_n)\,h + b(\widehat{S}_n)\,Z_n\,\sqrt{h}$$

if $\Delta_n = \dfrac{\partial \widehat{S}_n}{\partial S_0}$, then $\Delta_{n+1} = D_n\,\Delta_n, \quad D_n \equiv \dfrac{\partial F_n(\widehat{S}_n)}{\partial \widehat{S}_n}$,
and hence

$$\frac{\partial f(\widehat{S}_N)}{\partial S_0} = \frac{\partial f(\widehat{S}_N)}{\partial \widehat{S}_N}\Delta_N = \frac{\partial f}{\partial S}\,D_{N-1}\,D_{N-2}\ldots\,D_0\,\Delta_0$$

Generic adjoint approach

If S is m-dimensional, then D_n is an $m\times m$ matrix, and the computational cost per timestep is $O(m^3)$.

Alternatively,

$$\frac{\partial f(\widehat{S}_N)}{\partial S_0} = \frac{\partial f}{\partial S}\,D_{N-1}\,D_{N-2}\cdots D_0\,\Delta_0 = V_0^T\Delta_0,$$

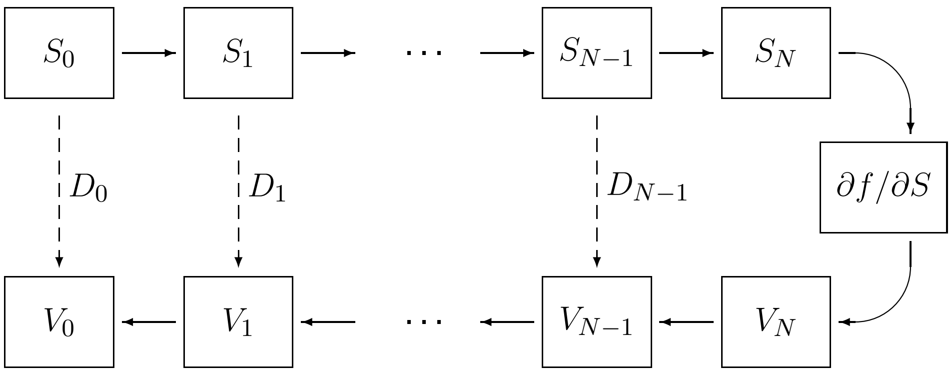
where adjoint $V_n = \left(\dfrac{\partial f(\widehat{S}_N)}{\partial \widehat{S}_n}\right)^T$ is calculated from

$$V_n = D_n^T V_{n+1}, \quad V_N = \left(\frac{\partial g}{\partial \widehat{S}_N}\right)^T,$$

at a computational cost which is $O(m^2)$ per timestep.

Generic adjoint approach

Note the flow of data within the path calculation:



– memory requirements are not significant because data only needs to be stored for the current path.

Generic adjoint approach

To calculate the sensitivity to other parameters, consider a generic parameter θ. Defining $\Theta_n = \partial\widehat{S}_n/\partial\theta$, then

$$\Theta_{n+1} = \frac{\partial F_n}{\partial S}\,\Theta_n + \frac{\partial F_n}{\partial\theta} \equiv D_n\,\Theta_n + B_n,$$

and hence

$$\begin{aligned}\frac{\partial f}{\partial\theta} &= \frac{\partial f}{\partial \widehat{S}_N}\,\Theta_N \\ &= \frac{\partial f}{\partial \widehat{S}_N}\Big\{B_{N-1} + D_{N-1}B_{N-2} + \ldots \\ &\qquad\qquad + D_{N-1}D_{N-2}\ldots D_1 B_0\Big\} \\ &= \sum_{n=0}^{N-1} V_{n+1}^T B_n.\end{aligned}$$

Generic adjoint approach

Different θ's have different B's, but same V's

$$\Longrightarrow \text{ Computational cost} \simeq m^2 \; + \; m \times \#\text{ parameters},$$

compared to the standard forward approach for which

$$\text{Computational cost} \simeq m^2 \times \#\text{ parameters}.$$

However, the adjoint approach only gives the sensitivity of one output, whereas the forward approach can give the sensitivities of multiple outputs for little additional cost.

LMM implementation

The generic description shows the potential for significant savings, but real implementations can exploit features of specific applications for additional savings.

For the LIBOR market model, this gives a factor m savings:

Cost per timestep	Value	forward Deltas	adjoint Deltas
Generic	$O(m^2)$	$O(m^3)$	$O(m^2)$
Optimized	$O(m)$	$O(m^2)$	$O(m)$

LMM implementation

Working through the details of the adjoint formulation, one eventually finds that $V_{i,n} = V_{i,n+1}$ for $i < \eta(nh)$, and

$$V_{i,n} = \frac{L_{i,n+1}}{L_{i,n}} V_{i,n+1} \; + \; \frac{\sigma_i^T \delta_i \, h}{(1+\delta_i L_{i,n})^2} \sum_{j=i}^{m} L_{j,n+1} V_{j,n+1} \sigma_j$$

for $i \geq \eta(nh)$.

Each timestep, there is an $O(m)$ cost in computing the summations, and then an $O(m)$ cost in updating the V_i's.

The correctness of the formulation is verified by checking it gives the same sensitivities as the forward calculation.

Generic adjoint approach

- This LMM example is not "lucky": an efficient evaluation $\Longrightarrow$ an efficient adjoint implementation;
- This is guaranteed by a theoretical result from the field of *algorithmic differentiation* which applies generic ideas to computer codes;
- Going further, *automatic differentiation* tools generate new computer code to perform standard (forward mode) and adjoint (reverse mode) sensitivity calculations;
- For more information see `http://www.autodiff.org`

Conclusions

- Greeks are vital for hedging and risk analysis
- Finite difference approximation is simplest to implement, but far from ideal
- Likelihood ratio method good for discontinuous payoffs
- In all other cases, pathwise sensitivities are best
- Payoff regularization (i.e. smoothing) may handle the problem of discontinuous payoffs
- Adjoint pathwise approach gives an unlimited number of sensitivities for a cost comparable to the initial valuation