Stochastic Simulation: Lecture 8

Prof. Mike Giles

Oxford University Mathematical Institute

Quasi-Monte Carlo

As in lecture 3, quasi-Monte Carlo methods can offer much greater accuracy for the same computational costs.

Same ingredients:

- Sobol or lattice rule quasi-uniform generators
- PCA to best use QMC inputs for multi-dimensional applications
- randomised QMC to regain confidence interval

New ingredient:

- how best to use QMC inputs to generate Brownian increments

Quasi-Monte Carlo

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

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

Quasi-Monte Carlo

The task now is to find a matrix L such that

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

We will consider 2 possibilities:

- Cholesky factorisation
- Brownian Bridge treatment

Cholesky factorisation

The Cholesky factorisation gives

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

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

Brownian Bridge construction

The "Brownian bridge" construction uses the following bit of theory:

If $t_1 < t < t_2$, then the distribution of $W(t)$, conditional on the values of $W(t_1)$ and $W(t_2)$, is

$$N\left(s\,W(t_1) + (s_1 - s)W(t_2),\ s(1-s)(t_2 - t_1)\right)$$

where $s = (t - t_1)/(t_2 - t_1)$.

Brownian Bridge construction

Using this, if the number of timestep M is a power of 2 then 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

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

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.

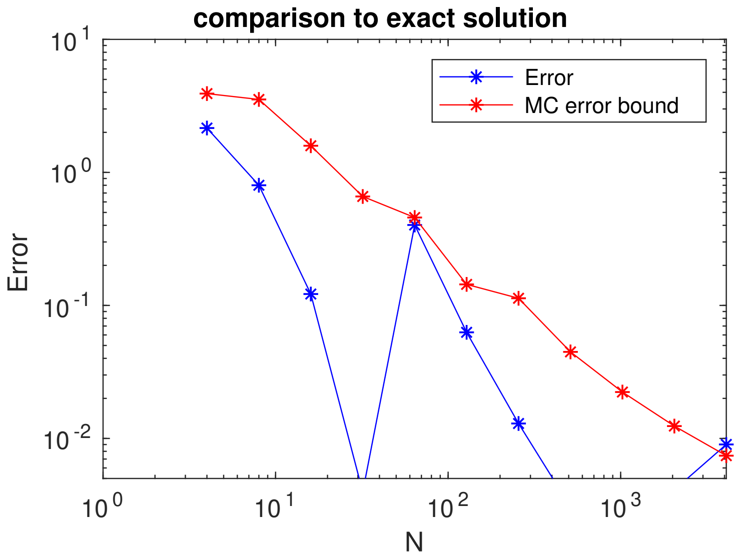
(This assumes M is a power of 2 – if not, the implementation is slightly more complex)

Numerical results

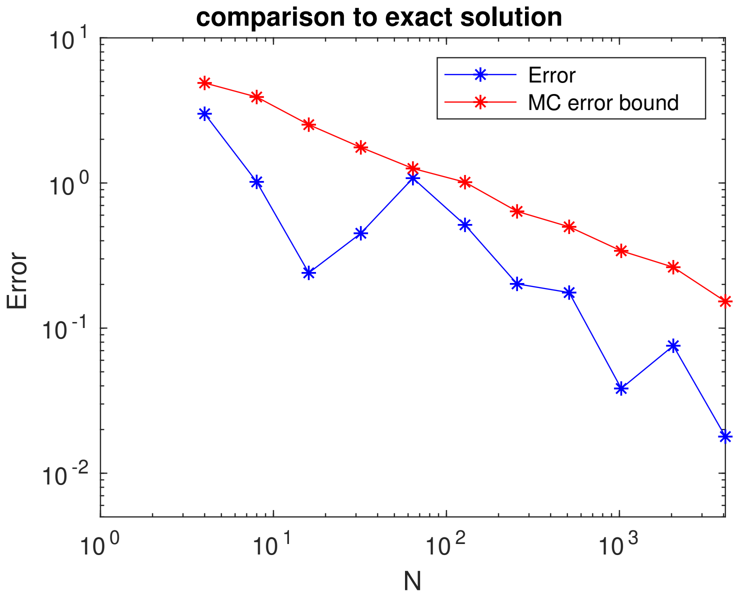
Usual European call test case based on geometric Brownian motion:

- 128 timesteps so weak error is negligible
- comparison between
  - QMC using Brownian Bridge
  - QMC without Brownian Bridge
  - standard MC
- QMC calculations use Sobol generator
- 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

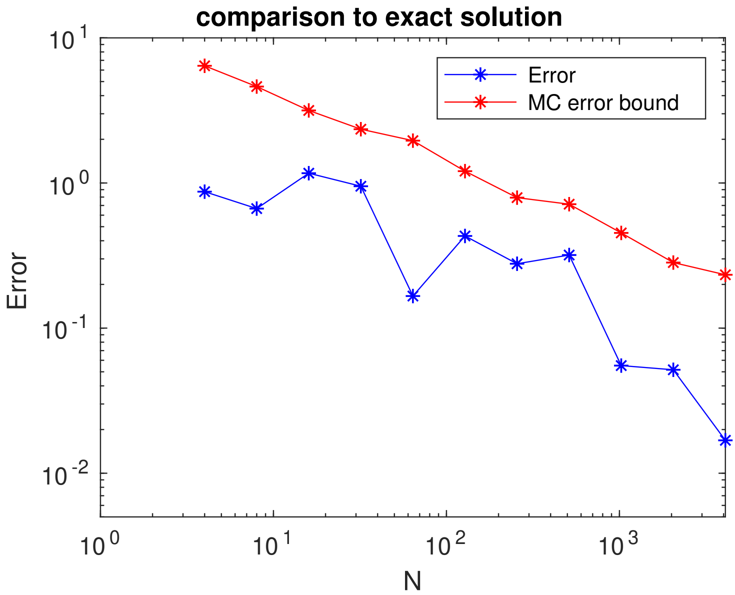
QMC with Brownian Bridge



QMC without Brownian Bridge



Standard Monte Carlo



QMC with Brownian Bridge

Why is QMC with Brownian Bridge so good?

For Geometric Brownian Motion, the final value S_T depends only only W_T, not on the rest of the Brownian path, so the Brownian Bridge construction reduces things to a 1-dimensional problem, dependent only on the first component Z_1.

QMC is extremely good for 1-dimensional problems, so the error is roughly $O(1/N)$.

For more general SDEs and almost all path-dependent option functions it is still the case that this reduces the effective dimensionality improving the effectiveness of QMC.

Multilevel Path Simulation

With SDEs, level ℓ corresponds to approximation using M^ℓ timesteps, giving approximate payoff $\widehat{P}_\ell$ at cost $C_\ell = O(M^\ell)$.

Simplest estimator for $\mathbb{E}[\widehat{P}_\ell - \widehat{P}_{\ell-1}]$ for $\ell > 0$ is

$$\widehat{Y}_\ell = N_\ell^{-1} \sum_{n=1}^{N_\ell} \left(\widehat{P}_\ell^{(n)} - \widehat{P}_{\ell-1}^{(n)} \right)$$

using same driving Brownian path for both levels.

Multilevel Path Simulation

Due to $O(h^{1/2})$ strong convergence,

$$\mathbb{E}[\,(\widehat{X}_{\ell,T} - X_T)^2] = O(h_\ell) \quad\Longrightarrow\quad \mathbb{E}[(\widehat{X}_{\ell,T} - \widehat{X}_{\ell-1,T})^2] = O(h_\ell)$$

so for Lipschitz payoff functions $P \equiv f(X_T)$, we have

$$\begin{aligned} V_\ell \;\equiv\; \mathbb{V}\left[\widehat{P}_\ell - \widehat{P}_{\ell-1}\right] &\leq \mathbb{E}\left[(\widehat{P}_\ell - \widehat{P}_{\ell-1})^2\right] \\ &\leq K^2\,\mathbb{E}\left[(\widehat{X}_{T,\ell} - \widehat{X}_{T,\ell-1})^2\right] \\ &= O(h_\ell) \end{aligned}$$

Also, due to weak convergence,

$$\mathbb{E}[\widehat{P}_\ell - P] = O(h_\ell).$$

Multilevel Path Simulation

In terms of the MLMC theorem, this means we have

$$\begin{aligned} C_\ell = O(M^\ell) &\implies \gamma = \log_2 M, \\ V_\ell = O(h_\ell) = O(M^{-\ell}) &\implies \beta = \log_2 M, \\ \mathbb{E}[\widehat{P}_\ell - P] = O(h_\ell) = O(M^{-\ell}) &\implies \alpha = \log_2 M, \end{aligned}$$

and therefore the overall cost to achieve ε RMS accuracy is $O(\varepsilon^{-2}|\log \varepsilon|^2)$.

Multilevel Path Simulation

The implementation is quite straightforward.

For each fine path timestep, we simulate the Brownian increment $\Delta W_n \sim N(0, h)$.

For a coarse timestep of size $M\, h$ we simply sum the M corresponding fine path increments to obtain the corresponding coarse path Brownian increment ΔW, and use this.

MLMC SDE algorithm

Input: fine and coarse timesteps h^f, h^c, final time $T = N\, h^c$, refinement factor $M = h^c/h^f$, initial states $\widehat{X}^f = \widehat{X}^c = X$

for $n = 1, N$ **do**

$\quad \Delta W^c := 0$

$\quad$ **for** $m = 1, M$ **do**

$\quad\quad$ generate r.v. $\Delta W^f \sim N(0, h^f)$

$\quad\quad \Delta W^c := \Delta W^c + \Delta W^f$

$\quad\quad \widehat{X}^f := \widehat{X}^f + a(\widehat{X}^f)\, h^f + b(\widehat{X}^f)\, \Delta W^f$

$\quad$ **end for**

$\quad \widehat{X}^c := \widehat{X}^c + a(\widehat{X}^c)\, h^c + b(\widehat{X}^c)\, \Delta W^c$

end for

$\widehat{P}_\ell - \widehat{P}_{\ell-1} := f(\widehat{X}^f) - f(\widehat{X}^c)$

MLMC extra bits – discontinuous functions

If the terminal function $f(S)$ is discontinuous at K then, heuristically,

- $O(h^{1/2})$ difference between $\widehat{X}^f$ and $\widehat{X}^c$
- $O(h^{1/2})$ probability of $\widehat{X}^f$ being within $O(h^{1/2})$ of K
- $\implies O(h^{1/2})$ probability of $f(\widehat{X}^f) - f(\widehat{X}^c) = O(1)$
- $\mathbb{E}[\,(\widehat{P}_\ell - \widehat{P}_{\ell-1})^2] = O(h^{1/2})$
- $\implies \alpha = \log_2 M,\ \beta = \frac{1}{2}\log_2 M,\ \gamma = \log_2 M$
- Overall complexity is $O(\varepsilon^{-5/2})$

This argument can be made rigorous – leads to $\mathbb{E}[\,(\widehat{P}_\ell - \widehat{P}_{\ell-1})^2] = O(h^{1/2-\delta})$ and overall complexity $O(\varepsilon^{-5/2-\delta})$ for any $\delta > 0$.

MLMC extra bits – Milstein

Milstein discretisation gives $O(h)$ strong convergence and hence

- $O(h^2)$ variance for Lipschitz $f(S_T)$
- $O(h^2)$ variance for function $f(\overline{S})$ based on path average
- With careful treatment, $O(h^2|\log h|^2)$ variance for $f(S)$ which is Lipschitz function of S_T and path minimum or maximum
- With careful treatment, $O(h^{3/2-\delta})$ variance for f which is discontinuous function of S_T or path minimum or maximum
- In all cases, sufficient for $O(\varepsilon^{-2})$ complexity

MLMC extra bits – adaptive time-stepping

Adaptive time-stepping perfectly within MLMC, again using the same Brownian motion for coarse and fine paths.

$\Delta W^c := 0,\ \Delta W^f := 0,\ t := 0,\ t^f := h^f,\ t^c := h^c$
while $\min(t^f, t^c) < T$ **do**
  generate r.v. $\Delta W \sim N(0, \min(t^f, t^c) - t)$
  $\Delta W^f := \Delta W^f + \Delta W, \quad \Delta W^c := \Delta W^c + \Delta W$
  $t := \min(t^f, t^c)$
  if $t^f = t$ **then**
    $\widehat{X}^f := \widehat{X}^f + a(\widehat{X}^f)\, h^f + b(\widehat{X}^f)\, \Delta W^f$
    calculate h^f, $\Delta W^f := 0,\ t^f := t^f + h^f$
  end if
  if $t^c = t$ **then**
    $\widehat{X}^c := \widehat{X}^c + a(\widehat{X}^c)\, h^c + b(\widehat{X}^c)\, \Delta W^c$
    calculate h^c, $\Delta W^c := 0,\ t^c := t^c + h^c$
  end if
end while

MLMC extra bits – other work

- MLQMC for SDEs – G, Waterhouse (2009)
- financial sensitivities ("Greeks") – Burgos (2011)
- American options – Belomestny & Schoenmakers (2011)
- jump-diffusion models – G, Xia (2012)
- Lévy-driven processes – Dereich (2010), Marxen (2010), Dereich & Heidenreich (2011), Kyprianou (2014)
- multi-dim. Milstein without Lévy areas – G, Szpruch (2014)
- expected exit times – Higham *et al* (2013), G, Bernal (2018)
- adaptive timesteps – Hoel, von Schwerin, Szepessy, Tempone (2012), G, Lester, Whittle (2014), Fang, G (2018, 2019)
- exponential Lévy processes – Xia (2017),
- reflected diffusions – Katsiolides *et al* (2018), G, Ramanan

Key references

P. Glasserman. "Monte Carlo Methods in Financial Engineering". Springer, 2003.

M.B. Giles. "Multilevel Monte Carlo path simulation" . Operations Research, 56(3):607-617, 2008.

M.B. Giles. "Improved multilevel Monte Carlo convergence using the Milstein scheme". pp.343-358, in Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer, 2008.

At least 80 articles listed in
`http://people.maths.ox.ac.uk/gilesm/mlmc_community.html`