Multilevel Monte Carlo methods using approximate distributions

Mike Giles
Oliver Sheridan-Methven

Mathematical Institute, University of Oxford

CERFACS seminar, April 14, 2021

(based on talk for 28th Biennial Conference on Numerical Analysis, June 28, 2019)

Overview

- Multilevel Monte Carlo (MLMC)
- approximate Normal random variables
- nested MLMC
- outline of numerical analysis
- numerical results
- conclusions / future work
- (a few extra slides)

Monte Carlo

Monte Carlo simulation uses the average

$$N^{-1} \sum_{n=1}^{N} P^{(n)}$$

to estimate $\mathbb{E}[P]$.

To achieve a RMS error of ε requires $N \approx \varepsilon^{-2} V$ samples, where $V = \mathbb{V}[P]$ is the variance.

If each sample costs C then the total cost is approximately $\varepsilon^{-2} VC$

Two-level Monte Carlo

If $\widetilde{P} \approx P$, then since $\mathbb{E}[P] = \mathbb{E}[\widetilde{P}] + \mathbb{E}[P - \widetilde{P}]$ we can instead use

$$N_0^{-1} \sum_{n=1}^{N_0} \widetilde{P}^{(n)} + N_1^{-1} \sum_{n=1}^{N_1} (P^{(n)} - \widetilde{P}^{(n)}).$$

The cost of this estimator is $N_0 C_0 + N_1 C_1$, and the variance is $V_0/N_0 + V_1/N_1$, where $V_0 \equiv \mathbb{V}[\widetilde{P}]$, $V_1 \equiv \mathbb{V}[P - \widetilde{P}]$.

Minimising the cost subject to the same accuracy requirement gives the total cost

$$\varepsilon^{-2} (\sqrt{V_0 C_0} + \sqrt{V_1 C_1})^2.$$

If

$$C_0 = 10^{-1} C, \quad C_1 = C, \quad V_0 = V, \quad V_1 = 10^{-3} V$$

then the total cost is $0.121\, \varepsilon^{-2} VC$, a factor 8 savings compared to the original Monte Carlo.

Multilevel Monte Carlo

Given a sequence of increasingly accurate (and costly) approximations $\widehat{P}_0, \widehat{P}_1, \widehat{P}_2, \ldots \to P$, for example from the approximation of an SDE using 2^ℓ timesteps on level ℓ, then

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

and so the MLMC estimate is

$$N_0^{-1} \sum_{n=1}^{N_0} \widehat{P}_0^{(n)} + \sum_{\ell=1}^{L} N_\ell^{-1} \sum_{n=1}^{N_\ell} (\widehat{P}_\ell^{(\ell,n)} - \widehat{P}_{\ell-1}^{(\ell,n)}),$$

and the total cost ends up being approximately

$$\varepsilon^{-2} \left(\sum_{\ell=0}^{L} \sqrt{V_\ell C_\ell} \right)^2$$

MLMC Theorem

(Slight generalisation of version in 2008 *Operations Research* paper)

If there exist independent estimators $\widehat{Y}_\ell$ based on N_ℓ Monte Carlo samples, each costing C_ℓ, and positive constants $\alpha, \beta, \gamma, c_1, c_2, c_3$ such that $\alpha \geq \frac{1}{2}\min(\beta, \gamma)$ and

i) $\left|\mathbb{E}[\widehat{P}_\ell - P]\right| \leq c_1\, 2^{-\alpha\,\ell}$

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

iii) $\mathbb{V}[\widehat{Y}_\ell] \leq c_2\, N_\ell^{-1} 2^{-\beta\,\ell}$

iv) $\mathbb{E}[C_\ell] \leq c_3\, 2^{\gamma\,\ell}$

MLMC Theorem

then there exists a positive constant c_4 such that for any $\varepsilon < 1$ there exist L and N_ℓ for which the multilevel estimator

$$\widehat{Y} = \sum_{\ell=0}^{L} \widehat{Y}_\ell,$$

has a mean-square-error with bound $\mathbb{E}\left[\left(\widehat{Y} - \mathbb{E}[P]\right)^2\right] < \varepsilon^2$

with an expected computational cost C with bound

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

Approximate random variables

In some applications, generating the random numbers can be a significant cost, especially with QMC when inverting the CDF.

e.g. Poisson distribution, increments of a Lévy process, non-central chi-squared distribution (CIR model in finance)

Even with Normal random variables, cost of conversion from uniform r.v. to Normal is non-trivial for vector implementation.

This has led to previous research by Hefter, Mayer, Ritter and others.

Note: one way of generating a sample of a scalar random variable X with CDF $C(x)$, is to first generate a $(0,1)$ uniform random variable, then define $X = C^{-1}(U)$

Approximate random variables

Simplest example: Euler-Maruyama approximation of a scalar SDE:

$$\widehat{X}_{t_{m+1}} = \widehat{X}_{t_m} + a(\widehat{X}_{t_m})\, h + b(\widehat{X}_{t_m})\sqrt{h}\, Z_m$$

where Z_m is a unit Normal r.v. generated as

$$Z_m = \Phi^{-1}(U_m)$$

where U_m is a uniform $(0,1)$ r.v. and Φ^{-1} is the inverse Normal CDF.

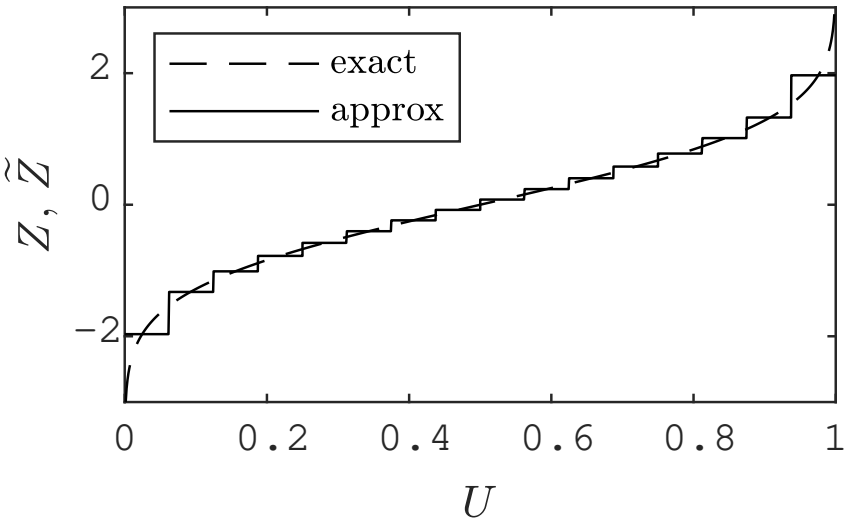
Suppose instead we use approximate Normals $\widetilde{Z}_m$ generated by

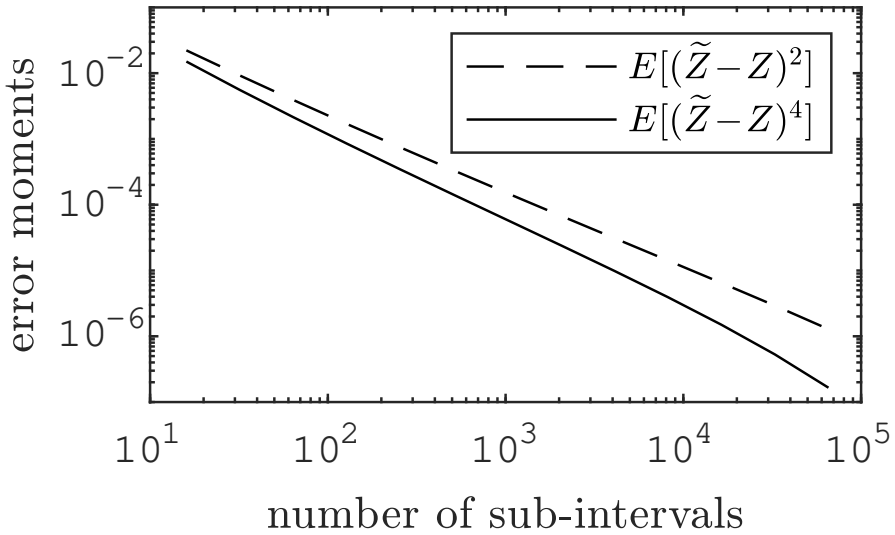
$$\widetilde{Z}_m = \widetilde{Q}(U_m)$$

where $\widetilde{Q}$ is an approximation to Φ^{-1}.

Approximate random variables

a) quantised approximation – piecewise constant on 2^d intervals

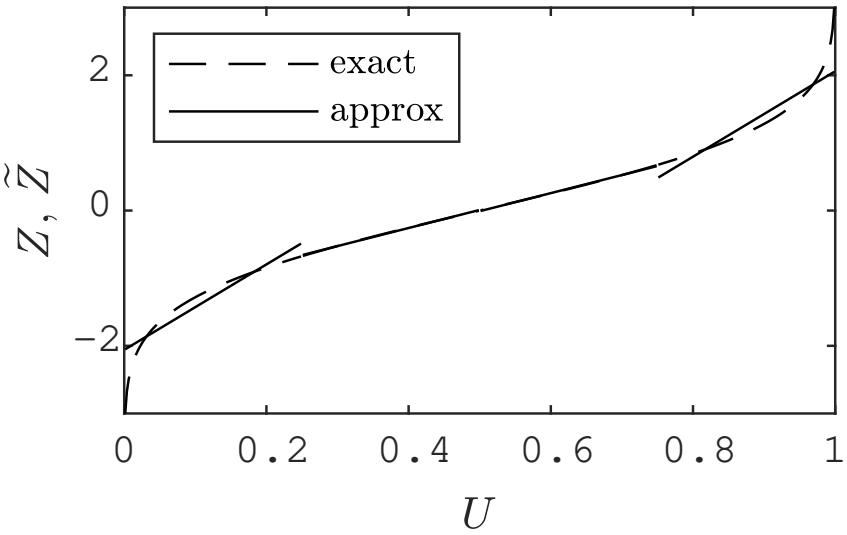


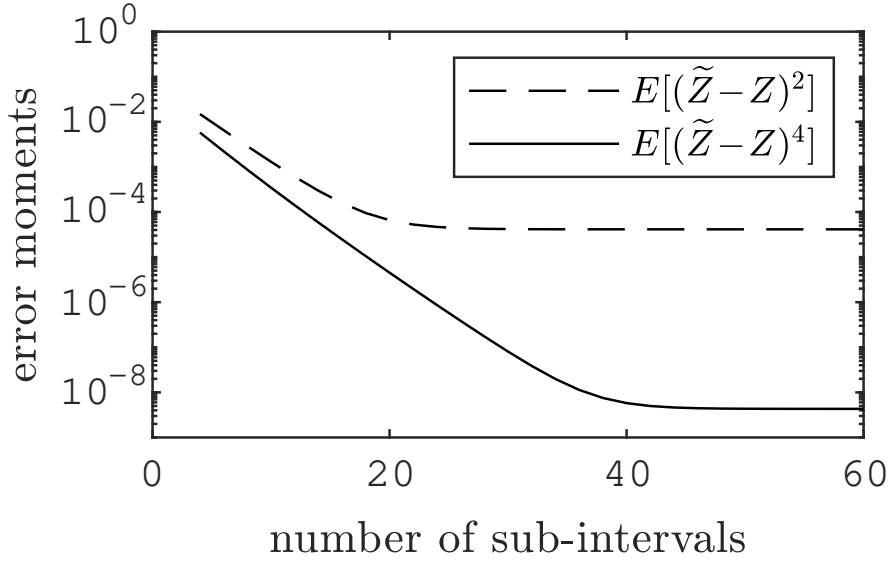


Good for scalar execution on CPUs – lookup in L1 cache based on d-digit random integer (d=10 is a reasonable choice)

Approximate random variables

b) piecewise linear on dyadic intervals – smaller near singularities



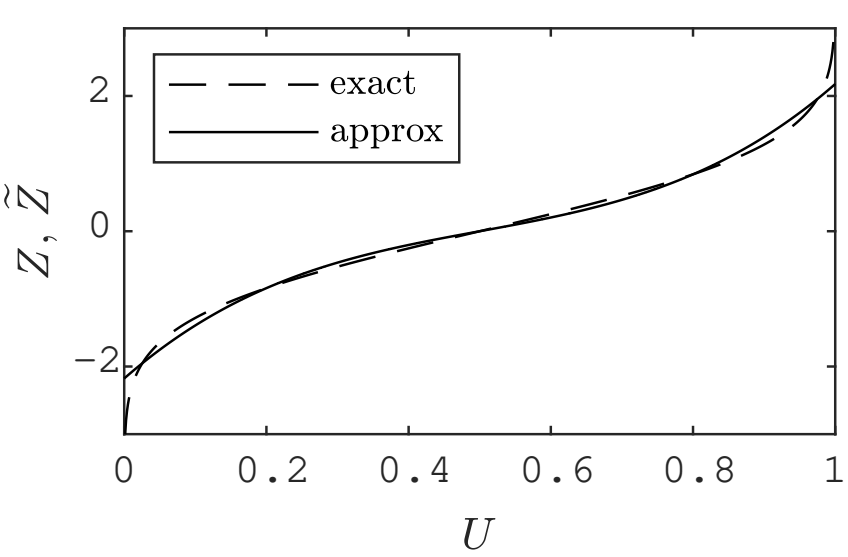


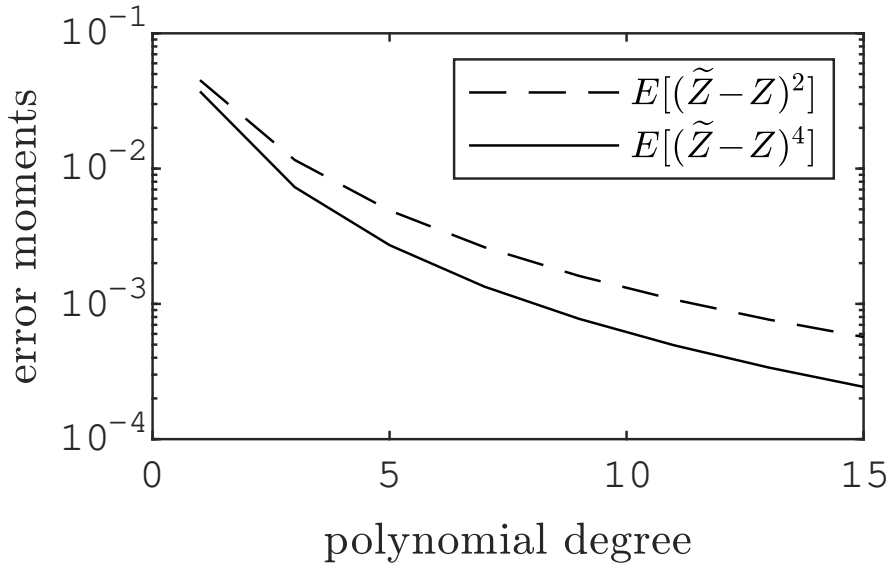
Good for vector execution on CPUs – lookup within a vector when 32 sub-intervals (cubic version used by Intel in their MKL software?)

Interval index from exponent of uniform random number in $(0, 1/2)$.

Approximate random variables

c) classic least-squares polynomial approximation





Good for half-precision on GPUs – no lookup
(degree 7 polynomial a reasonable approximation?)

Approximate random variables

For an SDE approximation with a specified number of timesteps, a two-level MLMC approach gives

$$\mathbb{E}[\widehat{P}] \;=\; \mathbb{E}[\widetilde{P}] \;+\; \mathbb{E}[\widehat{P}-\widetilde{P}]$$

Further analysis proves that for $P \equiv f(X_T)$ for a Lipschitz $f(x)$

$$\mathbb{V}[\widehat{P}-\widetilde{P}] = O\left(\mathbb{V}[Z-\widetilde{Z}]\right)$$

so this can lead to significant savings if we have both

- $\mathbb{V}[Z-\widetilde{Z}] \ll 1$
- $\widetilde{Z}_m \equiv \widetilde{Q}(U_m)$ is much cheaper to evaluate than $Z_m \equiv \Phi^{-1}(U_m)$

Approximate random variables

How does this work in combination with standard timestepping MLMC?

Answer: nested MLMC

$$\begin{aligned}
\mathbb{E}[\widehat{P}_L] &= \mathbb{E}[\widehat{P}_0] + \sum_{\ell=1}^{L} \mathbb{E}[\widehat{P}_\ell - \widehat{P}_{\ell-1}] \\
&= \mathbb{E}[\widetilde{P}_0] + \mathbb{E}[\widehat{P}_0 - \widetilde{P}_0] \\
&+ \sum_{\ell=1}^{L} \left\{ \mathbb{E}[\widetilde{P}_\ell - \widetilde{P}_{\ell-1}] + \mathbb{E}\left[(\widehat{P}_\ell - \widehat{P}_{\ell-1}) - (\widetilde{P}_\ell - \widetilde{P}_{\ell-1}) \right] \right\}
\end{aligned}$$

The pair $(\widetilde{P}_\ell, \widetilde{P}_{\ell-1})$ are generated in the same way as $(\widehat{P}_\ell, \widehat{P}_{\ell-1})$, just replacing exact Z_m by approximate $\widetilde{Z}_m$, for same underlying U_m

Approximate random variables

Numerical analysis – a nice challenge!

Path differences?

$$\begin{aligned}
\text{(strong convergence)} \qquad & \widehat{X}_\ell - \widehat{X}_{\ell-1} \;\sim\; h^{1/2} \\
\text{(similar, but non-standard)} \qquad & \widetilde{X}_\ell - \widetilde{X}_{\ell-1} \;\sim\; h^{1/2} \\
\text{(similar, but non-standard)} \qquad & \widehat{X}_\ell - \widetilde{X}_\ell \;\sim\; \sqrt{\mathbb{V}[Z-\widetilde{Z}]} \\
\text{(tricky!)} \qquad & (\widehat{X}_\ell - \widehat{X}_{\ell-1}) - (\widetilde{X}_\ell - \widetilde{X}_{\ell-1}) \;\sim\; h^{1/2}\sqrt{\mathbb{V}[Z-\widetilde{Z}]}
\end{aligned}$$

More precisely, for any $q > 2$ there is a constant c s.t.

$$\sqrt{\mathbb{E}\left[\,\Big((\widehat{X}_\ell - \widehat{X}_{\ell-1}) - (\widetilde{X}_\ell - \widetilde{X}_{\ell-1})\Big)^2\right]} \le c\, h^{1/2} \Big(\mathbb{E}[\,|\widetilde{Z}-Z|^q]\Big)^{1/q}$$

Approximate random variables

MLMC variances?

For smooth payoff functions $f(x)$, can prove

$$\mathbb{V}[(\widehat{P}_\ell - \widehat{P}_{\ell-1}) - (\widetilde{P}_\ell - \widetilde{P}_{\ell-1})] \sim h \left(\mathbb{E}[\,|\widetilde{Z} - Z|^q]\right)^{2/q}$$

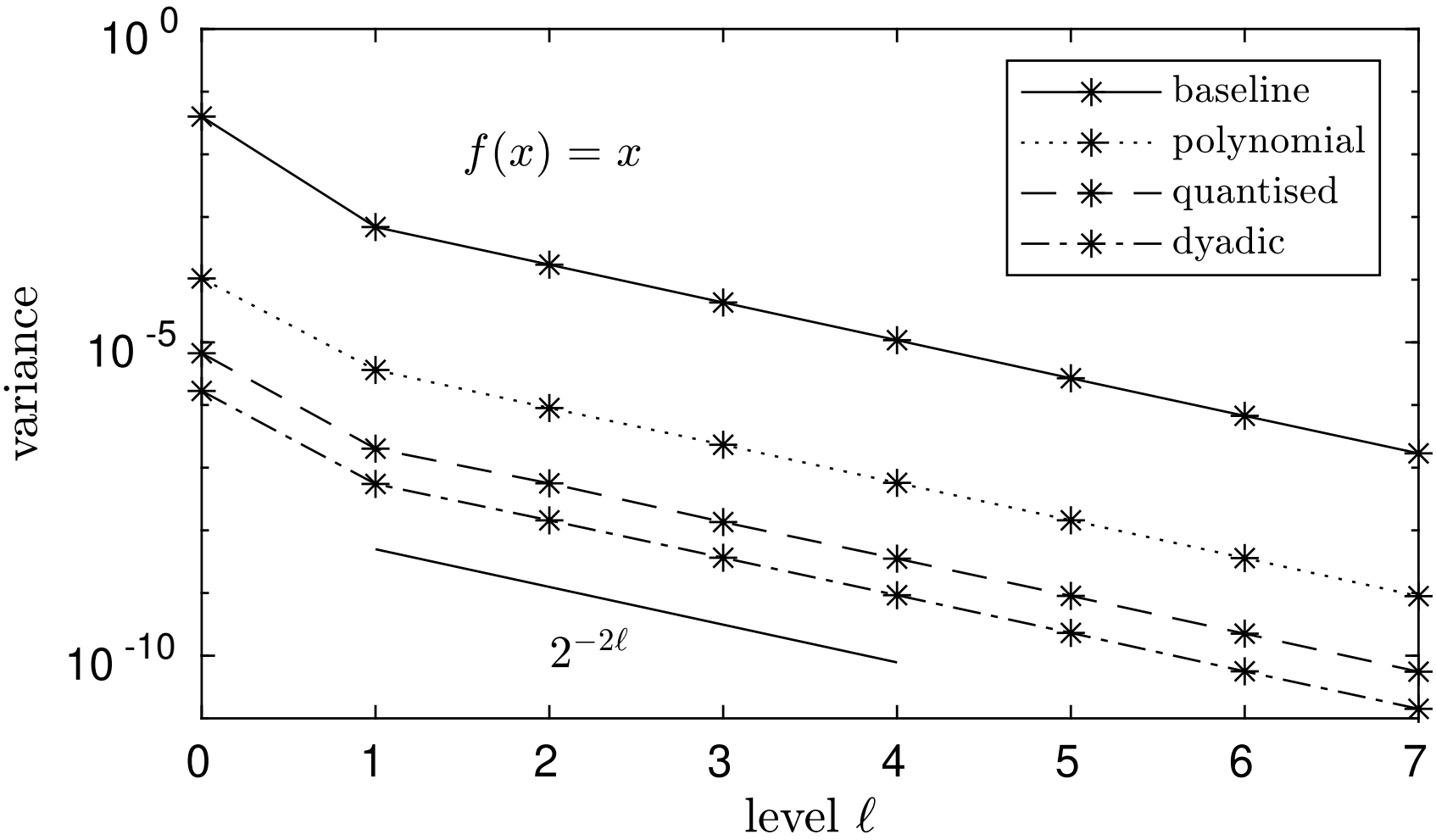
For continuous piecewise smooth functions (e.g. financial put/call functions) the analysis is much tougher. Numerical results suggest

$$\mathbb{V}[(\widehat{P}_\ell - \widehat{P}_{\ell-1}) - (\widetilde{P}_\ell - \widetilde{P}_{\ell-1})] \sim \min\left\{h^{1/2}\,\mathbb{V}[Z - \widetilde{Z}], h\sqrt{\mathbb{V}[Z - \widetilde{Z}]}\right\}$$

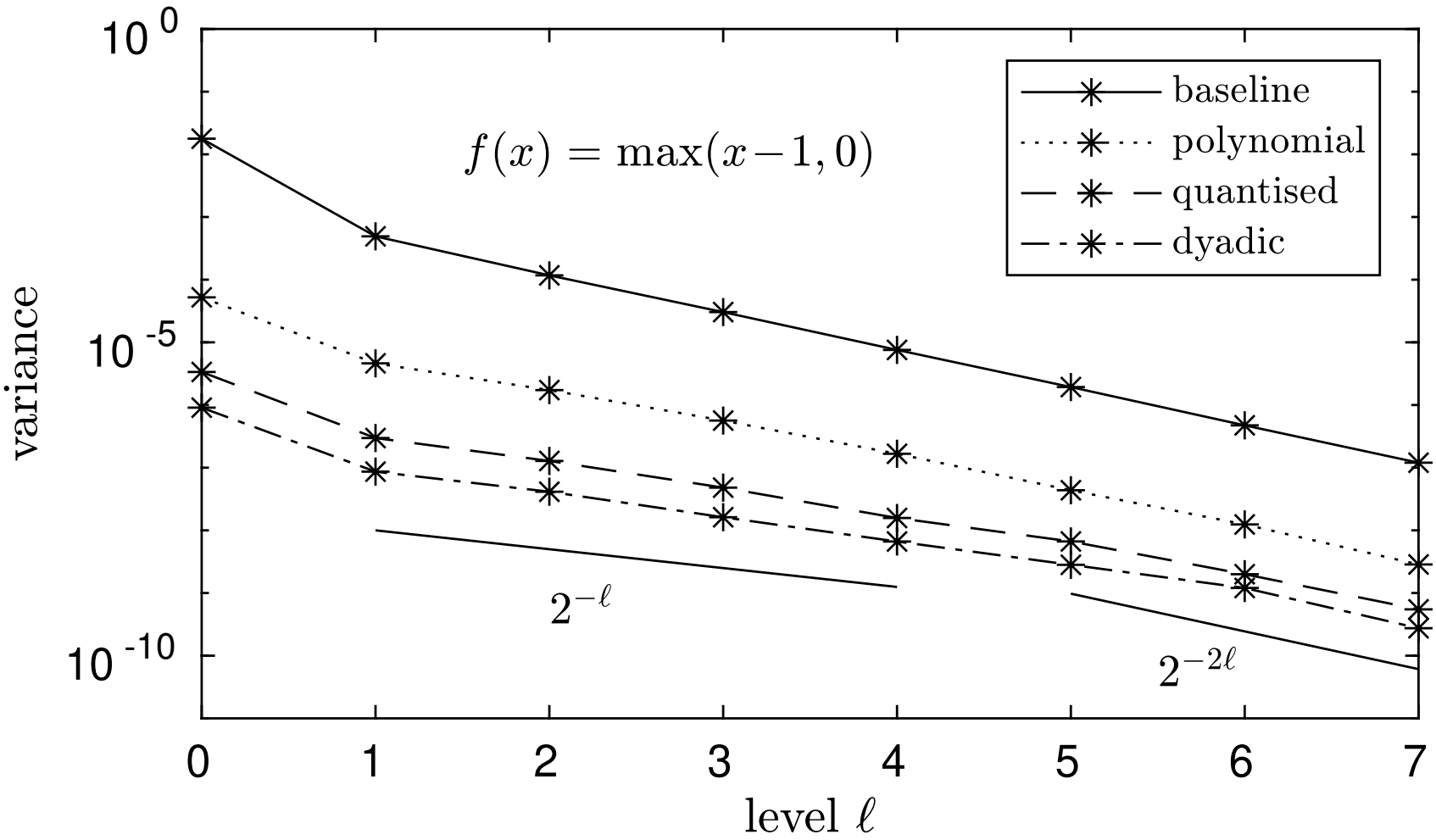
but unable to prove better than for any $q > 2$ and $\delta > 0$

$$\mathbb{V}[(\widehat{P}_\ell - \widehat{P}_{\ell-1}) - (\widetilde{P}_\ell - \widetilde{P}_{\ell-1})] \sim \min\left\{h^{\frac{1-\delta}{2} - \frac{1}{q}}\,\mathbb{E}[\,|\widetilde{Z} - Z|^q]^{\frac{2}{q}}, h\,\mathbb{E}[\,|\widetilde{Z} - Z|^q]^{\frac{1-\delta}{q+1}}\right\}$$

Numerical results



Numerical results



Reduced precision arithmetic

Further computational savings can be achieved by using reduced precision arithmetic.

Previous research at TU Kaiserslautern (Korn, Ritter, Wehn and others) and Imperial College (Luk and others) has used FPGAs with complete control over the precision, but we prefer GPUs.

In the latest NVIDIA GPUs, half-precision fp16 is twice as fast as single precision fp32 (which is 2-8 times faster than double precision fp64).

In most cases, single precision is perfectly sufficient for calculating $\widehat{P}$; half precision can be used for $\widetilde{P}$.

MC averaging should probably be done in double precision in both cases.

Reduced precision arithmetic

Very important: to ensure the telescoping sum is respected, must ensure that **exactly** the same computations are performed for $\widetilde{P}$ whether on its own or in calculating $\widehat{P} - \widetilde{P}$.

The effect of half-precision arithmetic can be modelled as

$$\widetilde{X}_{t_{m+1}} = \widetilde{X}_{t_m} + a(\widetilde{X}_{t_m})h + b(\widetilde{X}_{t_m})\sqrt{h}\,\widetilde{Z}_m + \delta\,\widetilde{X}_{t_m} V_m$$

where $\delta \approx 10^{-3}$ and the V_m are iid unit variance random variables.

Overall, this leads to an $O(\delta^2/h)$ increase in the variance; if this increases it too much, the reduced precision should not be used.

Conclusions / future work

- nested MLMC is similar to MIMC, but more general
- helpful in using approximate distributions and reduced precision
- offers significant computational savings in some situations
- in future, perform experiments with reduced precision, and extend to other distributions:
  - Poisson
  - binomial
  - non-central chi-squared (CIR process)
- also want to extend to MLQMC using quasi-uniform random numbers

References

MBG, M. Hefter, L. Mayer, K. Ritter. 'Random bit quadrature and approximation of distributions on Hilbert spaces'. *Foundations of Computational Mathematics*, 19(1):205-238, 2019.

MBG, M. Hefter, L. Mayer, K. Ritter. 'Random bit multilevel algorithms for stochastic differential equations'. *Journal of Complexity*, 54:101395, 2019.

MBG, M. Hefter, L. Mayer, K. Ritter. 'An adaptive random bit multilevel algorithm for SDEs'. In *Multivariate Algorithms and Information-Based Complexity*, De Gruyter, 2020.

MBG, O. Sheridan-Methven. 'Analysis of nested multilevel Monte Carlo using approximate Normal random variables'. arXiv pre-print, 2021.

O. Sheridan-Methven. 'Nested multilevel Monte Carlo methods and a modified Euler-Maruyama scheme utilising approximate Gaussian random variables suitable for vectorised hardware and low-precisions'. PhD thesis, University of Oxford, 2021.

Generation of uniform random variables

Randomised QMC often generates points as

$$U^{(m,n)} = S^{(m)} \wedge R^{(n)}$$

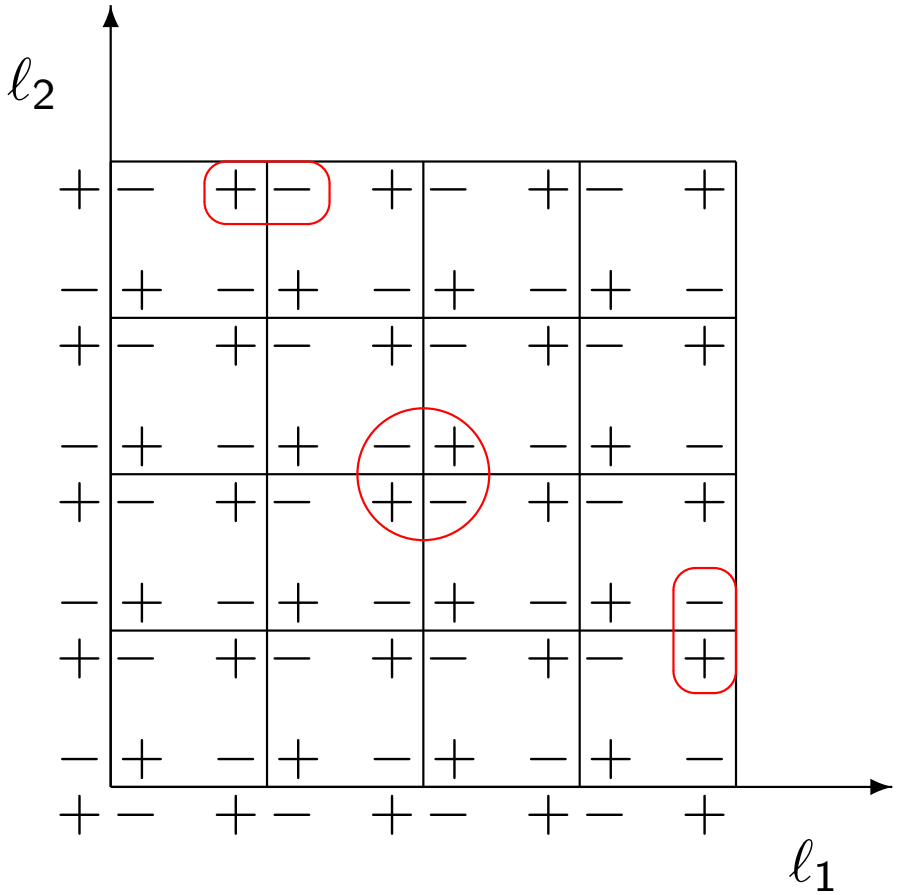
where $S^{(m)}$ is a set of Sobol points in $[0,1)^d$, $R^{(n)}$ is a set of pseudo-random points in $[0,1)^d$, and $\wedge$ is a bitwise exclusive-OR operator.

The same approach works for $S^{(m)}$ being pseudo-random points, and gives a point set $\{U^{(m,n)}\}$ with pairwise independence, which is all that is needed for standard variance analysis.

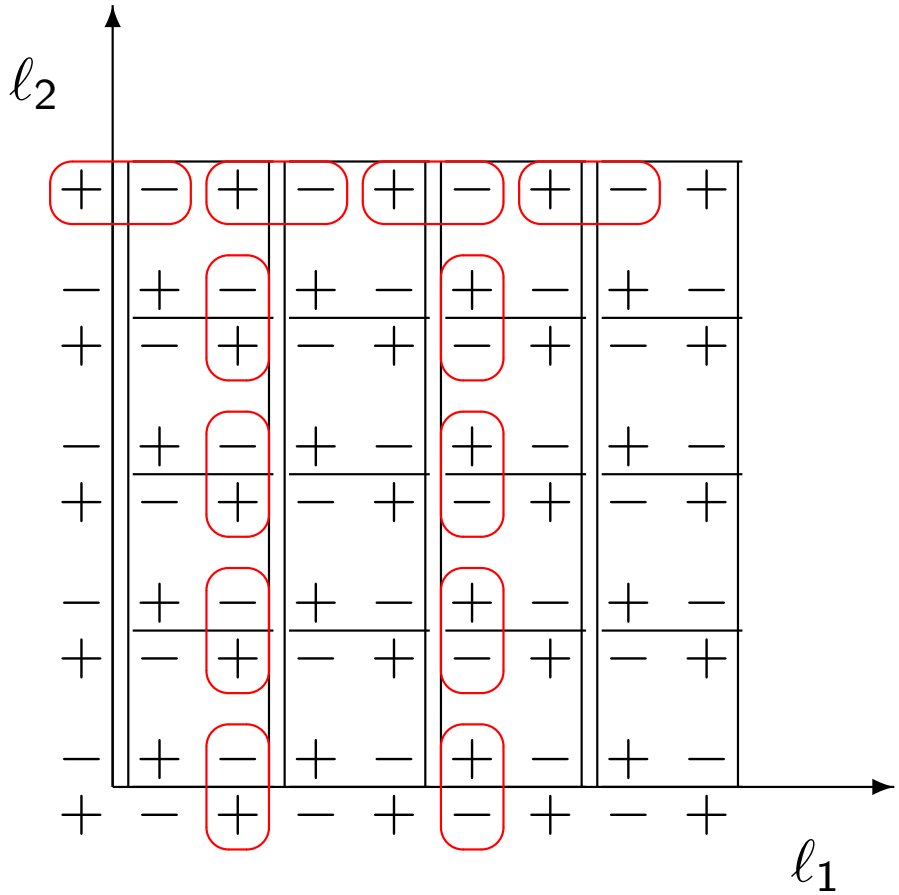
This effectively eliminates the cost of uniform random number generation – two sets of 1000 random numbers give rise to 10^6 random numbers at an average cost of less than 2 operations each.

MIMC compared to nested MLMC

MIMC: 4-way cancellation



nested MLMC: 2-way cancellation



MIMC compared to nested MLMC

A key aspect of the standard MLMC for SDEs is that the Brownian increment for a timestep of $2h$ is equal to the sum of Brownian increments for two timesteps of size h.

Another way of putting this is that if Z_1, Z_2 are unit Normal r.v.'s, then

$$\sqrt{h}\ Z_1 + \sqrt{h}\ Z_2 = \sqrt{2h}\ Z_3$$

where Z_3 is also a unit Normal r.v.

However, if $\widetilde{Z}_1, \widetilde{Z}_2$ are approximate unit Normal random variables, and

$$\sqrt{h}\ \widetilde{Z}_1 + \sqrt{h}\ \widetilde{Z}_2 = \sqrt{2h}\ \widetilde{Z}_3$$

then $\widetilde{Z}_3$ is an approximate Normal r.v. from a different distribution.

This is why we use the nested MLMC approach and not MIMC.