Multilevel Monte Carlo methods

Mike Giles

mike.giles@maths.ox.ac.uk

Oxford-Man Institute of Quantitative Finance

Mathematical Institute, University of Oxford

Sylvestre Burgos, Christoph Reisinger, Lukas Szpruch, Yuan Xia (Oxford)

Des Higham, Xuerong Mao (Strathclyde)

Frances Kuo, Ian Sloan, Ben Waterhouse (UNSW)

Rob Scheichl, Aretha Teckentrup, Elisabeth Ullmann (Bath)

Andrew Cliffe, Minho Park (Nottingham)

Kristian Debrabant (Southern Denmark), Andreas Rößler (Darmstadt)

Objectives

In presenting the multilevel Monte Carlo method, I hope to emphasise:

- the simplicity of the idea
- its flexibility
- that it's not prescriptive, more an approach
- scope for improved performance through being creative
- a growing number of people working on a variety of applications

I will focus on ideas rather than lots of numerical results.

Control variate

Classic approach to variance reduction: approximate $\mathbb{E}[f]$ using

$$N^{-1} \sum_{n=1}^{N} \left\{ f^{(n)} - \lambda \left(g^{(n)} - \mathbb{E}[g] \right) \right\}$$

where

- control variate g has known expectation $\mathbb{E}[g]$
- g is well correlated with f, and optimal value for λ can be estimated by a few samples

Two-level Monte Carlo

If we want to estimate $\mathbb{E}[f_1]$ but it is much cheaper to simulate $f_0 \approx f_1$, then since

$$\mathbb{E}[f_1] = \mathbb{E}[f_0] + \mathbb{E}[f_1 - f_0]$$

we can use the estimator

$$N_0^{-1} \sum_{n=1}^{N_0} f_0^{(n)} \; + \; N_1^{-1} \sum_{n=1}^{N_1} \left(f_1^{(n)} - f_0^{(n)} \right)$$

Two differences from standard control variate method:

- $\mathbb{E}[f_0]$ is not known, has to be estimated
- $\lambda = 1$

Two-level Monte Carlo

If we define

- C_0, V_0 to be cost and variance of f_0
- C_1, V_1 to be cost and variance of $f_1 - f_0$

then the total cost is

$$N_0\, C_0 \; + N_1\, C_1$$

and the variance (assuming independent estimators) is

$$N_0^{-1}\, V_0 \; + N_1^{-1}\, V_1$$

so for a fixed cost the variance is minimised by choosing

$$\frac{N_1}{N_0} = \frac{\sqrt{V_1/C_1}}{\sqrt{V_0/C_0}}$$

Trivial example

- f_1 comes from double precision calculation
- f_0 comes from single precision calculation (often twice as fast on latest CPUs/GPUs)
- use the same random number generator for both calculations

- estimating V_0 and V_1 will give an optimal allocation of computational effort between single precision and double precision computations

Multilevel Monte Carlo

Natural generalisation: given a sequence $f_0, f_1, \ldots, f_L$

$$\mathbb{E}[f_L] = \mathbb{E}[f_0] + \sum_{\ell=1}^{L} \mathbb{E}[f_\ell - f_{\ell-1}]$$

we can use the estimator

$$N_0^{-1} \sum_{n=1}^{N_0} f_0^{(n)} \; + \; \sum_{\ell=1}^{L} \left\{ N_\ell^{-1} \sum_{n=1}^{N_\ell} \left(f_\ell^{(n)} - f_{\ell-1}^{(n)} \right) \right\}$$

with independent estimation for each level

Multilevel Monte Carlo

If we define

- C_0, V_0 to be cost and variance of f_0
- C_ℓ, V_ℓ to be cost and variance of $f_\ell - f_{\ell-1}$

then the total cost is $\displaystyle\sum_{\ell=0}^{L} N_\ell \, C_\ell$

and the variance is $\displaystyle\sum_{\ell=0}^{L} N_\ell^{-1} \, V_\ell$

so for a fixed cost the variance is minimised by choosing

$$N_\ell \propto \sqrt{V_\ell / C_\ell} \quad \Longrightarrow \quad N_\ell \, C_\ell \propto \sqrt{V_\ell \, C_\ell}$$

Parametric Integration

Stefan Heinrich introduced multilevel ideas in 1999 for parametric integration, in which x is a finite-dimensional random variable, and want to estimate $\mathbb{E}[f(x,\lambda)]$ for a range of values of the parameter λ.

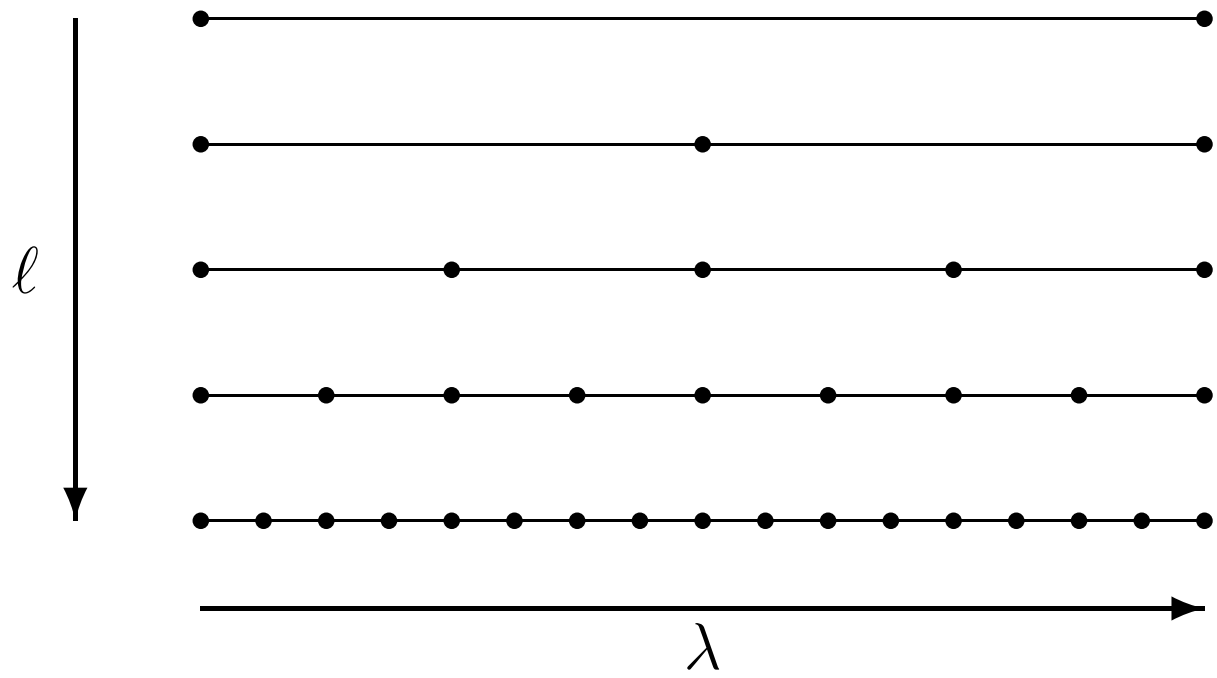
In the simplest case, suppose λ is a scalar, and the parameter range is $0 \le \lambda \le 1$.

If we have already estimated $\mathbb{E}[f(x,0)]$ and $\mathbb{E}[f(x,1)]$ then

$$\begin{aligned}\mathbb{E}[f(x,\tfrac{1}{2})] &= \tfrac{1}{2}\Big(\mathbb{E}[f(x,0)] + \mathbb{E}[f(x,1)]\Big) \\ &\quad + \mathbb{E}\left[f(x,\tfrac{1}{2}) - \tfrac{1}{2}(f(x,0)+f(x,1))\right]\end{aligned}$$

Parametric Integration

This can be repeated on multiple levels (perhaps using higher order interpolation if $f(x, \lambda)$ is sufficiently smooth)



This doesn't quite fit into the multilevel framework I've described, but the complexity analysis is very similar.

Multilevel Path Simulation

In 2006, I introduced the multilevel approach for infinite-dimensional integration arising from scalar SDE driven by Brownian diffusion.

Level ℓ corresponds to approximation using 2^ℓ timesteps, giving approximate payoff $\widehat{P}_\ell$.

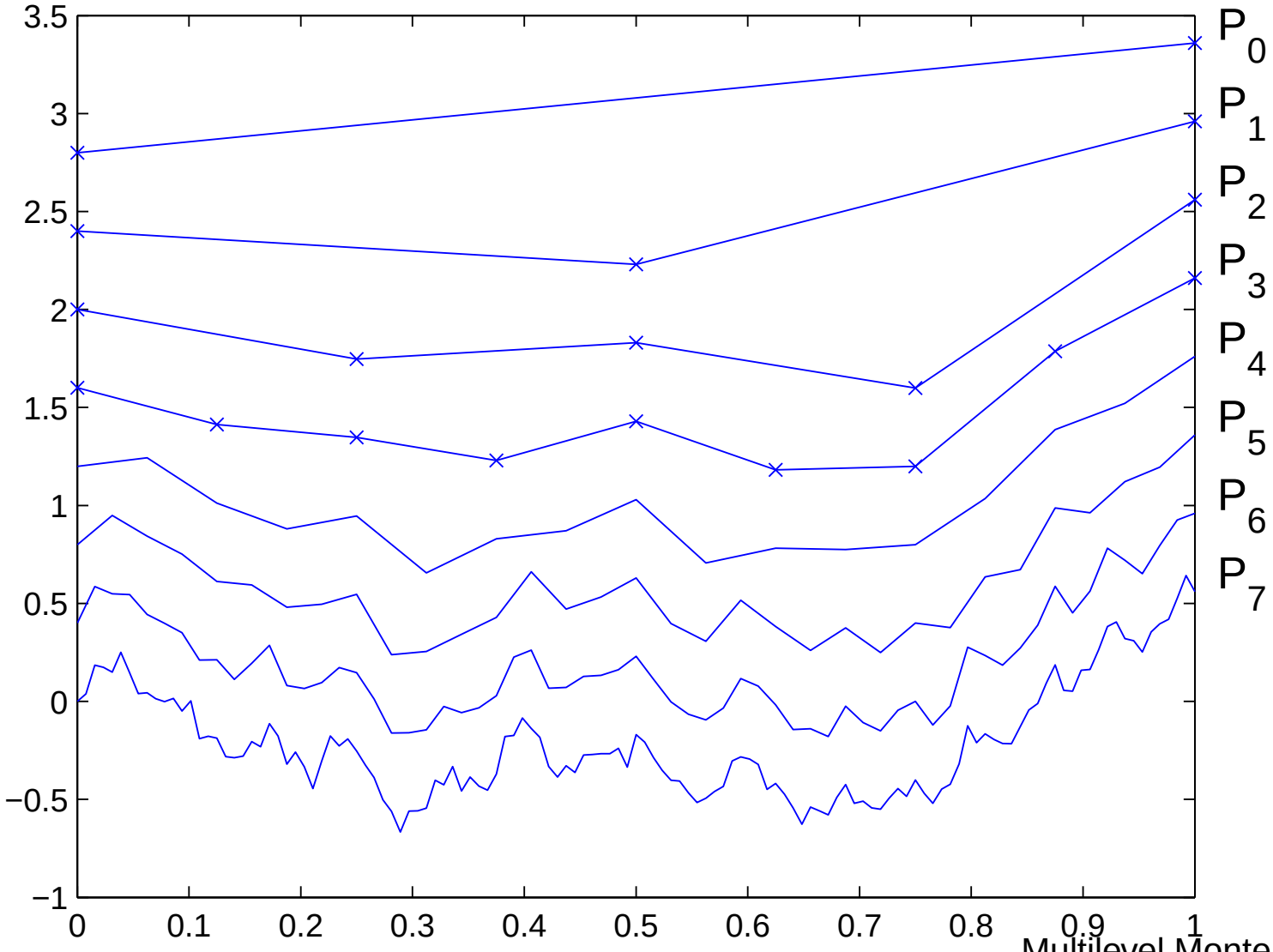
Choice of finest level L depends on weak error (bias).

Multilevel decomposition gives

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

Multilevel Monte Carlo

Discrete Brownian path at different levels



Multilevel Monte Carlo

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

Variance is $N_\ell^{-1} V_\ell$ where $V_\ell = \mathbb{V}[\widehat{P}_\ell - \widehat{P}_{\ell-1}]$ gets smaller as ℓ increases because $\widehat{P}_\ell, \widehat{P}_{\ell-1}$ both approximate same P

To make RMS error less than ε

- choose L so that $\left(\mathbb{E}[\widehat{P}_L] - \mathbb{E}[P] \right)^2 < \frac{1}{2} \varepsilon^2$
- choose $N_\ell \propto \sqrt{V_\ell / C_\ell}$ so total variance is less than $\frac{1}{2} \varepsilon^2$

MLMC Theorem

(Slight generalisation of original version)

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

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

iv) $C_\ell \leq c_3\, 2^{\gamma\,\ell}$

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 a computational cost C with bound

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

MLMC Theorem

Monte Carlo simulation requires $O(\varepsilon^{-2})$ samples to achieve RMS accuracy of ε.

MLMC theorem says that in the best case, in which the variance decays with level faster than the cost increases, the cost is optimal – $O(1)$ cost per sample, on average

To further reduce the overall cost would require switching to Multilevel QMC (more later on this)

MLMC Theorem

MLMC Theorem allows a lot of freedom in constructing the multilevel estimator. I sometimes use different approximations on the coarse and fine levels:

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

which is OK provided $\mathbb{E}[\widehat{P}_\ell^f(\omega^{(n)})] = \mathbb{E}[\widehat{P}_\ell^c(\omega^{(n)})]$

For example, could use $\widehat{P}_\ell^f(\omega^{(n)}) = \frac{1}{2}\left(\widehat{P}_\ell^c(\omega^{(n)}) + \widehat{P}_\ell^c(\omega_{anti}^{(n)}) \right)$ where $\omega_{anti}^{(n)}$ is an antithetic "twin" with the same distribution as $\omega^{(n)}$.

MLMC Challenges

- not always obvious how to couple coarse and fine levels
  i.e. what does $\widehat{P}_\ell(\omega^{(n)}) - \widehat{P}_{\ell-1}(\omega^{(n)})$ mean?
- can the MLMC flexibility be exploited to improve the variance decay?

  particularly important for discontinuous "payoffs", since a small difference in the coarse and fine "paths" can produce an $O(1)$ difference in the "payoff"
- numerical analysis – proving the rate at which V_ℓ decays can be tough

Brownian Diffusion SDEs

Brownian increments for coarse path obtained by summing increments for fine path – very simple and natural

I prefer to use the first order Milstein discretisation – for simple put / call options (and more generally Lipschitz functions of the final state) this leads to

$$\widehat{P}_\ell - \widehat{P}_{\ell-1} = O(h_\ell)$$

and hence $V_\ell = O(h_\ell^2)$.

However, it's not so easy for lookback and digital options.

(And in multiple dimensions, Milstein requires Lévy areas, but this can be avoided by an antithetic treatment, G & Szpruch, 2011)

Lookback options

Payoff depends on the minimum attained by the path $S(t)$.

If the numerical approximation uses the minimum of the values at the discrete simulation times

$$\widehat{S}_{min} \equiv \min_j \widehat{S}_j$$

then we have two problems:

- $O(\sqrt{h})$ weak convergence
- $\widehat{S}_{\ell,min} - \widehat{S}_{\ell-1,min} = O(\sqrt{h_\ell})$ which leads to $V_\ell = O(h_\ell)$

Lookback options

To fix this, define a Brownian Bridge interpolation conditional on the endpoints for each timestep, and approximating the drift and volatility as being constant.

For the fine path, we then use standard results for sampling from the distribution of the minimum of a Brownian Bridge to define

$$\widehat{S}_{min} = \min_j \ \frac{1}{2}\left(\widehat{S}_j + \widehat{S}_{j-1} - \sqrt{(\widehat{S}_j - \widehat{S}_{j-1})^2 - 2\,h\,b_j^2\log U_j}\right)$$

where the U_j are independent $U(0,1)$ random variables.

This gives $O(h)$ weak convergence, but if we do something similar for the coarse path with a different set of $U's$ the variance will still be poor.

Lookback options

Instead, do the following:

- sample from the mid-point of the Brownian Bridge interpolation for the coarse timestep, using the Brownian path information from the fine path – this mid-point value is within $O(h_\ell)$ of the fine path simulation
- sample from the minima of each half of the coarse timestep using the same $U's$ as fine path
- take the minimum of the two minima, and then the minimum over all coarse timesteps.

This leads to an $O(h_\ell)$ difference in the computed minima for the coarse and fine paths, and is valid because the distribution for the coarse path minimum has not been altered.

Digital options

Payoff is discontinuous function of the final state.

First order strong convergence means that $O(h_\ell)$ of the simulations have coarse and fine paths on opposite sides of a discontinuity.

Hence,

$$\widehat{P}_\ell - \widehat{P}_{\ell-1} = \begin{cases} O(1), & \text{with probability } O(h_\ell) \\ O(h_\ell), & \text{with probability } O(1) \end{cases}$$

so $V_\ell = O(h_\ell)$, not $O(h_\ell^2)$

Digital options

Three fixes:

- Splitting: split each path simulation into M paths by trying M different values for the Brownian increment for the last fine path timestep
- Conditional expectation: using the Euler discretisation instead of Milstein for the final timestep, conditional on all but the final Brownian increment the final state has a Gaussian distribution, with a known analytic conditional expectation in simple cases
- Change of measure: when the expectation is not known, can use a change of measure so the coarse path takes the same final state as the fine path – difference in the "payoff" now comes from the Radon-Nikodym derivative

Numerical Analysis

option	Euler		Milstein	
	numerics	analysis	numerics	analysis
Lipschitz	$O(h)$	$O(h)$	$O(h^2)$	$O(h^2)$
Asian	$O(h)$	$O(h)$	$O(h^2)$	$O(h^2)$
lookback	$O(h)$	$O(h)$	$O(h^2)$	$o(h^{2-\delta})$
barrier	$O(h^{1/2})$	$o(h^{1/2-\delta})$	$O(h^{3/2})$	$o(h^{3/2-\delta})$
digital	$O(h^{1/2})$	$O(h^{1/2}\log h)$	$O(h^{3/2})$	$o(h^{3/2-\delta})$

Table: V_l convergence observed numerically (for GBM) and proved analytically (for more general SDEs)

Euler analysis due to G, Higham & Mao (2009) and Avikainen (2009). Milstein analysis due to G, Debrabant & Rößler (2012).

Computing Greeks

(Sylvestre Burgos, 2011)

- MLMC combines well with pathwise sensitivity analysis for Greeks
- Main concern is reduced regularity of “payoff”
- Techniques are similar to handling digital options

Jump diffusion

Finite activity rate Merton-style jump diffusion
(Yian Xia, 2011)

- if fixed rate no problem, use jump-adapted discretisation and coarse and fine paths jump at the same time
- if path-dependent rate then it's trickier
  - use jump-adapted discretisation plus thinning (Glasserman & Merener)
  - could lead to fine and coarse paths jumping at different times – poor variance
  - instead use a change of measure to force jumps to be at the same time

Lévy processes

Infinite activity rate general Lévy processes
(Dereich 2010; Marxen 2010; Dereich & Heidenreich 2011)

- on level ℓ, simulate jumps bigger than δ_ℓ
  ($\delta_\ell \to 0$ as $\ell \to \infty$)
- either neglect smaller jumps or use a Gaussian approximation
- multilevel problem: how to handle jumps which are bigger than δ_ℓ but smaller than $\delta_{\ell-1}$?

Lévy processes

Exact simulation (Cheng Zhu, Filippo Zinzani)

- with some popular exponential-Lévy models (variance-gamma, NIG) possible to directly simulate increments over fine timesteps
- just sum them pairwise to get corresponding increments for coarse path
- coarse and fine path simulations are both exact, so what's the point of multilevel simulation?
  - Asian options
  - lookback options
  - barrier options
  - other path-dependent options

Heston stochastic volatility

Glasserman & Kim (2011) developed a series expansion for sampling from the integrated variance:

$$\left(\int_0^T V_s \, ds \;\middle|\; V_0 = v_0, V_t = v_t \right) \stackrel{d}{=} \sum_{n=1}^{\infty} x_n + \sum_{n=1}^{\infty} y_n + \sum_{n=1}^{\infty} z_n$$

where x_n, y_n, z_n are independent random variables.

Multilevel possibility:

- truncate series at K_ℓ $\quad (K_\ell \to \infty$ as $\ell \to \infty)$
- should help for European options as well as path-dependent options

American options

Belomestny & Schoenmakers (2011) have developed a multilevel implementation of upper bound dual pricing

- based on nested simulation algorithm of Andersen and Broadie (2004)
- requires sub-sampling at each timestep to estimate a conditional expectation (the continuation value)
- multilevel treatment uses a different number of sub-samples M_ℓ on each level ($M_\ell \to \infty$ as $\ell \to \infty$)

SPDEs

- very natural straightforward application, with better savings than SDEs due to higher dimensionality
- big challenge is in numerical analysis – noteworthy contribution by Charrier & Teckentrup (2010)
- range of applications
  - Hou (2007?) – elliptic
  - Graubner & Ritter (2008) – parabolic
  - Giles, Reisinger (2009-11) – parabolic
  - Barth, Lang, Schwab, Zollinger (2010/11) – elliptic, parabolic, hyperbolic
  - Cliffe, Giles, Scheichl, Teckentrup (2010/11) – elliptic

Elliptic SPDE

Elliptic PDE with random coefficient $k(\mathbf{x}, \omega)$:

$$-\nabla . \left(k(\mathbf{x}, \omega) \nabla p(\mathbf{x}, \omega)\right) = 0, \qquad \mathbf{x} \in D,$$

Model k as a lognormal random field, i.e. $\log k$ is a Gaussian field with mean 0 and covariance function

$$R(\mathbf{x}, \mathbf{y}) = \sigma^2 \exp\left(- \|\mathbf{x} - \mathbf{y}\|_1 / \lambda \right)$$

Samples of $\log k$ are provided by a Karhunen-Loève expansion:

$$\log k(\mathbf{x}, \omega) = \sum_{n=0}^{\infty} \sqrt{\theta_n}\ \xi_n(\omega)\ f_n(\mathbf{x}),$$

where ξ_n are iid unit Normal random variables.

Elliptic SPDE

In multilevel treatment:

- different spatial grid resolution on each level
- truncate KL-expansion at different cutoffs K_ℓ
- some benefit from using antithetic treatment:

$$\log k_{\ell-1}(\mathbf{x},\omega) = \sum_{n=0}^{K_{\ell-1}} \sqrt{\theta_n}\ \xi_n(\omega)\ f_n(\mathbf{x}),$$

$$\log k_\ell^+(\mathbf{x},\omega) \quad = \quad \log k_{\ell-1}(\mathbf{x},\omega) + \sum_{n=K_{\ell-1}+1}^{K_\ell} \sqrt{\theta_n}\ \xi_n(\omega)\ f_n(\mathbf{x}),$$

$$\log k_\ell^-(\mathbf{x},\omega) \quad = \quad \log k_{\ell-1}(\mathbf{x},\omega) - \sum_{n=K_{\ell-1}+1}^{K_\ell} \sqrt{\theta_n}\ \xi_n(\omega)\ f_n(\mathbf{x}),$$

Stochastic chemical reactions

In stochastic simulations, each reaction is a Poisson process with a rate which depends on the current concentrations.

In the “tau-leaping” method (Euler-Maruyama method) the reaction rates are frozen at the start of the timestep, then for each reaction sample from a Poisson process

$$P(\lambda \, \Delta t)$$

to determine the number of reactions in that timestep.

(As $\lambda \, \Delta t \to \infty$, the standard deviation becomes smaller relative to the mean, and it approaches the deterministic limit.)

Stochastic chemical reactions

Anderson & Higham (2011) have developed a very efficient multilevel version of this algorithm – big savings because finest level usually has 1000's of timesteps.

Key challenge: how to couple coarse and fine path simulations?

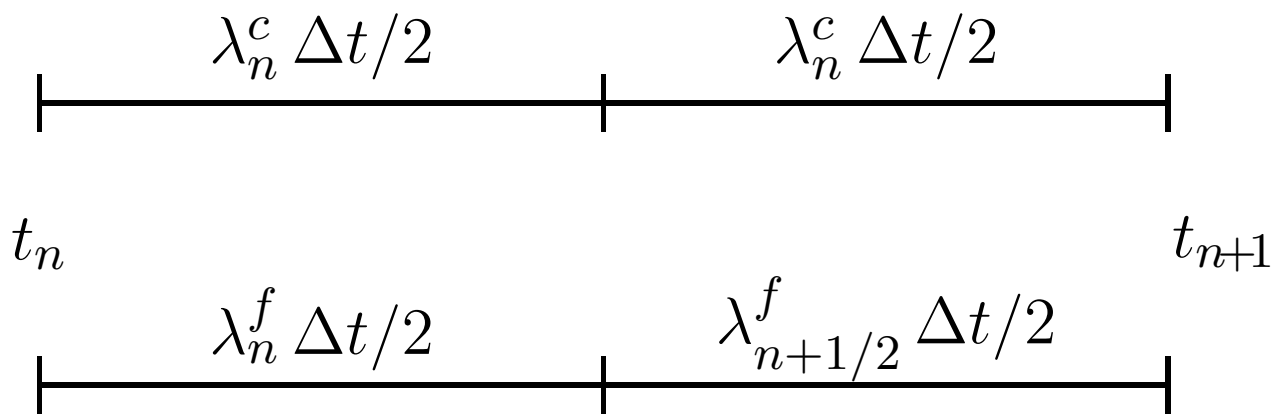
Crucial observation: $P(t_1) + P(t_2) \stackrel{d}{=} P(t_1+t_2)$

Only requirement: $t_1, t_2 \geq 0$

Stochastic chemical reactions

Solution:

- simulate the Poisson variable on the coarse timestep as the sum of two fine timestep Poisson variables
- couple the fine path and coarse path Poisson variables by using common variable based on smaller of two rates



If $\lambda_n^f < \lambda_n^c$, use $P(\lambda_n^c \Delta t/2) \sim P(\lambda_n^f \Delta t/2) + P((\lambda_n^c - \lambda_n^f)\Delta t/2)$

Wasserstein Metric

Sub-division of coarse path random variable into sum of two fine path random variables should work in many settings.

What about second part? Given two very similar scalar probability distributions, want to obtain samples Z^f, Z^c from each in a way which minimises $\mathbb{E}[\,|Z^f - Z^c|^p]$.

This corresponds to 1D version of Wasserstein metric:

$$\inf_{\gamma} \int \left\| Z^f - Z^c \right\|^p \, d\gamma(Z^f, Z^c)$$

where minimum is over all joint distributions with correct marginals.

Wasserstein Metric

In 1D, Wasserstein metric is equal to

$$\int_0^1 \left| \Phi_f^{-1}(u) - \Phi_c^{-1}(u) \right|^p \, \mathrm{d}u$$

where Φ_f and Φ_c are the cumulative probability distributions, and this minimum is achieved by choosing

$$Z^f = \Phi_f^{-1}(U), \quad Z^c = \Phi_c^{-1}(U),$$

for same uniform $[0, 1]$ random variable U.

This might be a good technique for future multilevel applications?

MLQMC

To further improve the multilevel complexity, can use randomised QMC in place of MC.

G & Waterhouse (2008-9) used rank-1 lattice rules for scalar SDE applications

- far fewer samples required on coarsest levels
- almost no difference on finest levels
- overall, big savings when using Milstein discretisation (so most work on coarsest levels)
- in best case (GBM with European option) complexity was approximately $O(\varepsilon^{-1.5})$

MLQMC

Numerical algorithm:

1. start with $L=0$
2. get an initial estimate for V_L using 32 random offsets and $N_L = 1$
3. while $\sum_{\ell=0}^{L} V_\ell > \varepsilon^2/2$, try to maximise variance reduction per unit cost by doubling N_ℓ on the level with largest $V_\ell \,/\, (C_\ell \, N_\ell)$
4. if $L<2$ or the bias estimate is greater than $\varepsilon/\sqrt{2}$, set $L := L+1$ and go back to step 2

Complexity

There are several papers now on the complexity of infinite-dimensional quadrature using multilevel approaches.

Paper by Müller-Gronbach & Ritter (2009) shows that under certain conditions, for a certain class of problems, multilevel methods are provably optimal, to within logarithmic terms.

Conclusions

- multilevel idea is very simple; key is thinking how to apply it in new situations
- lots of freedom to construct more efficient estimators using various “tricks”:
  - change of measure
  - antithetic treatment
  - sub-division
- being used for an increasingly wide range of applications; biggest computational savings when coarsest approximation is much cheaper than finest

Webpage for MLMC activity / papers:
`people.maths.ox.ac.uk/gilesm/mlmc_community.html`

MLMC Community

Abo Academi (Avikainen) – numerical analysis
Basel (Harbrecht) – elliptic SPDEs, sparse grid links
Bath (Scheichl, Teckentrup, Ullmann) – elliptic SPDEs
Christian-Albrechts University (Gnewuch) – multilevel QMC
Duisburg (Belomestny) – Bermudan and American options
ETH Zürich (Barth, Lang, Schwab) – SPDEs
Frankfurt (Gerstner, Kloeden) – numerical analysis, sparse grid links
IIT Chicago (Hickernell) – SDEs, infinite-dimensional integration, complexity analysis
Kaiserslautern (Heinrich, Korn, Neuenkirch, Ritter) – finance, SDEs, infinite-dimensional integration, complexity analysis, parametric integration
KAUST (Tempone, von Schwerin) – adaptive time-stepping
KTH (Szepessy) – adaptive time-stepping
Marburg (Dereich) – Lévy-driven SDEs
Munich (Hutzenthaler) – numerical analysis
Nottingham (Cliffe, Park) – elliptic SPDEs
Oxford (Giles, Reisinger, Szpruch) – SDEs, jump-diffusion, SPDEs, numerical analysis
Passau (Müller-Gronbach) – infinite-dimensional integration, complexity analysis
Princeton (Jentzen) – numerical analysis
Strathclyde (Higham, Mao) – numerical analysis, exit times, stochastic chemical modelling
WIAS (Schoenmakers) – Bermudan and American options
Wisconsin (Anderson) – numerical analysis, stochastic chemical modelling