Multilevel Monte Carlo methods

Mike Giles

Mathematical Institute, University of Oxford

SPPEXA Workshop on Numerical Methods
on High-Performance Computers

December 1, 2014

Computational Engineering

Initially, focus was on improving its analysis capability.

In CFD this meant:

- 1D $\rightarrow$ 2D $\rightarrow$ 3D
- steady $\rightarrow$ unsteady
- inviscid $\rightarrow$ inviscid+losses $\rightarrow$ viscous+turbulence modelling $\rightarrow$ large edge simulation $\rightarrow$ direct Navier-Stokes
- simple geometry $\rightarrow$ complex geometry (CAD, unstructured grids)
- coupling to combustion, heat transfer, aeroelastic effects

From an HPC perspective, this usually meant one big calculation using all of a supercomputer / cluster.

Computational Engineering

Next, the focus moved to design optimisation.

In CFD this meant:

- lots of calculations for optimisation based on genetic algorithms
- new adjoint techniques to reduce the cost for gradient-based optimisation when there are lots of design variables

In other areas there was also a move to multi-scale modelling – big applications with different parts interacting on different length scales

In HPC, this still usually meant one big calculation – genetic algorithms were used mainly on small problems

Computational Engineering

Now, I think the focus is on

- multi-physics coupling – linking together multiple big application codes in a way which remains scalable on big systems
- uncertainty quantification (UQ) – accounting for the uncertainty in
  - geometry (e.g. manufacturing tolerances)
  - model parameters (e.g. material properties)
  - initial conditions (e.g. weather prediction)
- robust multi-point design – changing the design process to account for uncertainty and use under different conditions

In HPC, this is increasingly meaning multiple calculations – means that individual applications don't have to scale to use the whole system

Uncertainty Quantification

Methods vary depending on:

- level of uncertainty
  - small, almost linear
  - large, definitely nonlinear
- dimensionality (number of uncertain parameters)

Main classes of method:

- moment methods – very effective for small uncertainties
- PDE methods (polynomial chaos, stochastic Galerkin, stochastic collocation) – very effective for low dimensions
- Monte Carlo methods – good for nonlinear high-dimensional problems

Monte Carlo

Monte Carlo is a very simple "brute force" method.

If we want to estimate the expected value (or average) $\mathbb{E}[f(\omega)]$ where ω comes from some input probability distribution, then use

$$\widehat{Y} = \frac{1}{N}\sum_{n=1}^{N} f(\omega^{(n)})$$

where $\omega^{(n)}$ are independent random samples

This is unbiased, $\mathbb{E}[\widehat{Y}] = \mathbb{E}[f]$, and has variance

$$\mathbb{V}[\widehat{Y}] \equiv \mathbb{E}[(\widehat{Y} - \mathbb{E}[f])^2] = N^{-1}\mathbb{V}[f]$$

so the r.m.s. error is $O(N^{-1/2})$.

Monte Carlo

The cost is proportional to N, so r.m.s. error $= O(\text{cost}^{-1/2})$

- good news: independent of "dimension" of uncertainty
- bad news: $O(\text{cost}^{-1/2})$ is a poor rate of decay – can need 1000's of calculations for reasonable accuracy

General perception – good simple method when calculations are cheap, but not when each calculation is expensive (e.g. needs a PDE solution)

Opinion is now changing:

- petascale/exascale computing offers more compute capability
- multilevel Monte Carlo reduces the cost significantly, especially for PDE applications

Objectives of talk

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
- lots of people working on a variety of applications

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

Control variate

There are lots of techniques to reduce the variance in Monte Carlo simulation, so fewer samples are needed for good accuracy.

One classic approach: 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

Using the optimal value for λ (which can be estimated) reduces the variance by factor $1-\rho^2$, where ρ is the correlation between f and g.

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, so 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 variance the cost 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

Less trivial example

- f_1 comes from simulation of Navier-Stokes equations with a turbulence model
- f_0 comes from simulation of Euler equations
- uncertainty in aircraft geometry due to manufacturing tolerances
- likely to work well at cruise conditions when Euler model is a reasonable approximation
- unlikely to work well under high-lift conditions when there are big flow separations

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

Using a Lagrange multiplier μ^2 to minimise the cost for a fixed variance

$$\frac{\partial}{\partial N_\ell} \sum_{k=0}^{L} \left(N_k \, C_k + \mu^2 N_k^{-1} V_k \right) = 0$$

gives

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

Multilevel Monte Carlo

Setting the total variance equal to ε^2 gives

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

and hence, the total cost is

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

in contrast to the standard cost which is approximately $\varepsilon^{-2} \, V_0 \, C_L$.

The MLMC cost savings are therefore:

- V_L/V_0, if $\sqrt{V_\ell C_\ell}$ increases with level
- C_0/C_L, if $\sqrt{V_\ell C_\ell}$ decreases with level

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.

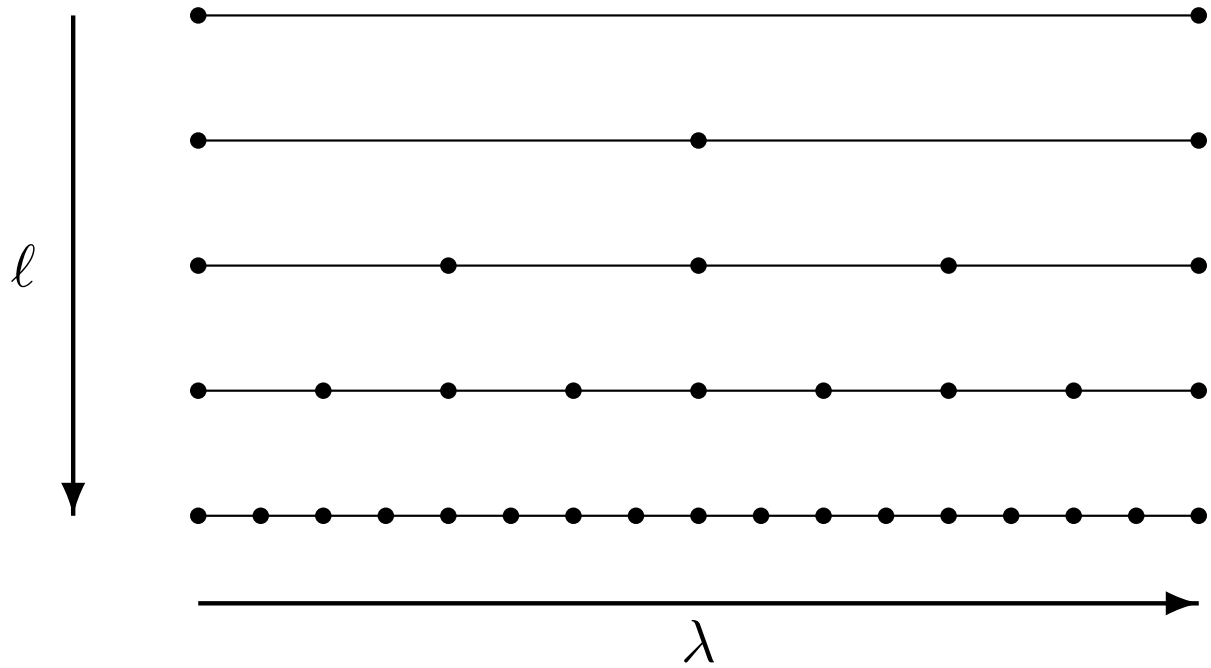
In the simplest case, suppose λ is a scalar, and the parameter range is $0 \le \lambda \le 1$.

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 SDEs driven by Brownian diffusion.

$$\mathrm{d}S_t = a(S_t, t)\,\mathrm{d}t + b(S_t, t)\,\mathrm{d}W_t$$

Here W_t is a Brownian path – increments $\mathrm{d}W_t$ are Normally distributed with zero mean and variance $\mathrm{d}t$.

Simple Euler-Maruyama approximation is

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

where ΔW_n is Normally-distributed with zero mean and variance Δt.

Used extensively in computational finance – often interested in $\mathbb{E}[P(S_T)]$.

Multilevel Path Simulation

Multilevel decomposition gives

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

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

Using same driving Brownian path for $\widehat{P}_\ell, \widehat{P}_{\ell-1}$ – this means summing the Brownian increments for pairs of fine path timesteps to get the Brownian increment for the coarse timestep

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

Multilevel Monte Carlo

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

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

with 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 (strong convergence)

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

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

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

MLMC Theorem

Two observations of optimality:

- MC simulation needs $O(\varepsilon^{-2})$ samples to achieve RMS accuracy ε. When $\beta > \gamma$, the cost is optimal — $O(1)$ cost per sample on average.

  (Would need multilevel QMC to further reduce costs)

- When $\beta < \gamma$, another interesting case is when $\beta = 2\alpha$, which corresponds to $\mathbb{E}[\widehat{Y}_\ell]$ and $\sqrt{\mathbb{E}[\widehat{Y}_\ell^2]}$ being of the same order as $\ell \to \infty$. In this case, the total cost is $O(\varepsilon^{-\gamma/\alpha})$, which is the cost of a single sample on the finest level — again optimal.

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?

- discontinuous output functionals are a problem, since a small difference between the coarse and fine solutions can produce a large difference in the output – needs some creative tricks

- numerical analysis – proving the rate at which V_ℓ decays can be tough

SPDEs

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

Engineering Uncertainty Quantification

- consider 3D elliptic PDE, with uncertain boundary data
- use grid spacing proportional to $2^{-\ell}$ on level ℓ
- cost is $O(2^{-3\ell})$, if using an efficient multigrid solver
- 2nd order accuracy means that

$$\widehat{P}_\ell(\omega) - \widehat{P}(\omega) \approx c(\omega)\, 2^{-2\ell}$$

$$\Longrightarrow \quad \widehat{P}_{\ell-1}(\omega) - \widehat{P}_\ell(\omega) \approx 3\, c(\omega)\, 2^{-2\ell}$$

- hence, $\alpha = 2,\ \beta = 4,\ \gamma = 3$
- cost is $O(\varepsilon^{-2})$ to obtain ε RMS accuracy

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

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

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

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

- (more efficient ways of generating $\log k_\ell$ use technique known as *circulant embedding*)

Reduced Basis PDE approximation

Vidal-Codina, Nguyen, G, Peraire (2014) take a very fine PDE discretisation:

$$A(\omega)\, u = f(\omega)$$

and use a reduced basis approximation

$$u \approx \sum_{k=1}^{K} v_k u_k$$

to generate a reduced system

$$A_r(\omega)\, v = f_r(\omega)$$

- larger $K \Longrightarrow$ greater accuracy at greater cost
- in multilevel treatment, K_ℓ varies with level

Iterative convergence

Most PDE solvers involve iterative solvers.

So far, have implicitly assumed we are converging the solution until the remaining error is negligible.

Alternatively, different levels in the multilevel formulation could use different numbers of iterations, or different convergence criteria.

No-one has tried this yet (as far as I know) – point here is that MLMC approach is very general and flexible, just need some hierarchy of approximations, with cost and accuracy increasing together

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, so 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: $P(t_1) + P(t_2) \overset{d}{=} P(t_1 + t_2)$

Only requirement: $t_1, t_2 \geq 0$

They used this to combine Poisson variates from two fine path timesteps to form a Poisson variate for a coarse timestep.

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
- in best case (GBM with European option) complexity was approximately $O(\varepsilon^{-1.5})$

Conclusions

- multilevel Monte Carlo idea is very simple
- being used for an increasingly wide range of applications; biggest computational savings when coarsest (helpful) approximation is much cheaper than finest
- currently, getting at least 100× savings for SPDEs and stochastic chemical reaction simulations
- webpage for my research/papers:
  `people.maths.ox.ac.uk/gilesm/mlmc.html`
- new *Acta Numerica* review article and codes:
  `people.maths.ox.ac.uk/gilesm/acta.html`

MLMC Community

Webpage: people.maths.ox.ac.uk/gilesm/mlmc_community.html

Abo Academi (Avikainen) – numerical analysis
Basel (Harbrecht) – elliptic SPDEs, sparse grid links
Bath (Kyprianou, Scheichl, Shardlow) – elliptic SPDEs, MCMC, Lévy-driven SDEs
Chalmers (Lang) – SPDEs
Christian-Albrechts University (Gnewuch) – multilevel QMC
Duisburg (Belomestny) – Bermudan and American options
Edinburgh (Davie, Szpruch) – SDEs, numerical analysis
ETH Zürich (Jenny, Jentzen, Schwab) – numerical analysis, SPDEs
Frankfurt (Gerstner, Kloeden) – numerical analysis, sparse grid links
Fraunhofer ITWM (Iliev) – SPDEs in engineering
Hong Kong (Chen) – Brownian meanders, nested simulation in finance
IIT Chicago (Hickernell) – SDEs, infinite-dimensional integration, complexity analysis
Kaiserslautern (Heinrich, Korn, Ritter) – finance, SDEs, complexity analysis, parametric integration
KAUST (Tempone) – adaptive time-stepping
Kiel (Gnewuch) – randomized multilevel QMC
Mannheim (Neuenkirch) – numerical analysis, fractional Brownian motion
Marburg (Dereich) – Lévy-driven SDEs
Munich (Hutzenthaler) – numerical analysis
Oxford (Giles, Hambly, Reisinger) – SDEs, jump-diffusion, SPDEs, numerical analysis
Passau (Müller-Gronbach) – infinite-dimensional integration, complexity analysis
Purdue (Gittelson) – SDPEs
Stanford (Glynn) – numerical analysis
Strathclyde (Higham, Mao) – numerical analysis, exit times, stochastic chemical modelling
Stuttgart (Barth) – SPDEs
Texas A&M (Efendiev) – SPDEs in engineering
UCLA (Caflisch) – Coulomb collisions in physics
UNSW (Dick, Kuo, Sloan) – multilevel QMC
WIAS (Schoenmakers) – Bermudan and American options
Wisconsin (Anderson) – numerical analysis, stochastic chemical modelling