Monte Carlo Methods for Uncertainty Quantification

Mike Giles

Mathematical Institute, University of Oxford

KU Leuven Summer School on Uncertainty Quantification

May 30–31, 2013

Lecture outline

Lecture 4: PDE applications

- PDEs with uncertainty
- examples
- multilevel Monte Carlo
- details of MATLAB code

PDEs with Uncertainty

Looking at the history of numerical methods for PDEs, the first steps were about improving the modelling:

- 1D $\rightarrow$ 2D $\rightarrow$ 3D
- steady $\rightarrow$ unsteady
- laminar flow $\rightarrow$ turbulence modelling $\rightarrow$ large eddy simulation $\rightarrow$ direct Navier-Stokes
- simple geometries (e.g. a wing) $\rightarrow$ complex geometries (e.g. an aircraft in landing configuration)
- adding new features such as combustion, coupling to structural / thermal analyses, etc.

. . . and then engineering switched from analysis to design.

PDEs with Uncertainty

The big move now is towards handling uncertainty:

- uncertainty in modelling parameters
- uncertainty in geometry
- uncertainty in initial conditions
- uncertainty in spatially-varying material properties
- inclusion of stochastic source terms

Engineering wants to move to "robust design" taking into account the effects of uncertainty.

Other areas want to move into Bayesian inference, starting with an *a priori* distribution for the uncertainty, and then using data to derive an improved *a posteriori* distribution.

PDEs with Uncertainty

Examples:

- Long-term climate modelling:

  Lots of sources of uncertainty including the effects of aerosols, clouds, carbon cycle, ocean circulation (`http://climate.nasa.gov/uncertainties`)

- Short-range weather prediction

  Considerable uncertainty in the initial data due to limited measurements

PDEs with Uncertainty

- Engineering analysis

  Perhaps the biggest uncertainty is geometric due to manufacturing tolerances

- Nuclear waste repository and oil reservoir modelling

  Considerable uncertainty about porosity of rock

- Astronomy

  "Random" spatial/temporal variations in air density disturb correlation in signals received by different antennas

- Finance

  Stochastic forcing due to market behaviour

PDEs with Uncertainty

In the past, Monte Carlo simulation has been viewed as impractical due to its expense, and so people have used other methods:

- stochastic collocation
- polynomial chaos

Because of Multilevel Monte Carlo, this is changing and there are now several research groups using MLMC for PDE applications

The approach is very simple, in principle:

- use a sequence of grids of increasing resolution in space (and time)
- as with SDEs, determine the optimal allocation of computational effort on the different levels
- the savings can be much greater because the cost goes up more rapidly with level

Generalised MLMC Theorem

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

Generalised 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

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

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

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

Parabolic SPDE

Unusual parabolic SPDE arises in a financial setting
(Bush, Hambly, Haworth & Reisinger)

$$\mathrm{d}p = -\mu \, \frac{\partial p}{\partial x} \, \mathrm{d}t + \frac{1}{2} \, \frac{\partial^2 p}{\partial x^2} \, \mathrm{d}t + \sqrt{\rho} \, \frac{\partial p}{\partial x} \, \mathrm{d}W$$

with absorbing boundary $p(0,t) = 0$

- derived in limit as number of firms $\longrightarrow \infty$
- x is distance to default
- $p(x,t)$ is probability density function
- stochastic $\mathrm{d}W$ term corresponds to systemic risk
  (a market crash or recession hits all firms)
- $\partial^2 p / \partial x^2$ comes from idiosyncratic risk
  (each firm is affected by its own unique circumstances)

Parabolic SPDE

Numerical discretisation combines Milstein time-marching with central difference spatial approximation:

$$
\begin{aligned}
v_j^{n+1} \;=\; v_j^n \;&-\; \frac{\mu\, k + \sqrt{\rho}\,\Delta W_n}{2h} \left(v_{j+1}^n - v_{j-1}^n\right) \\
&+\; \frac{(1-\rho)\, k + \rho\,\Delta W_n^2}{2h^2} \left(v_{j+1}^n - 2v_j^n + v_{j-1}^n\right)
\end{aligned}
$$

where k is the timestep, h is the grid spacing, and $\Delta W_n \sim N(0,k)$.

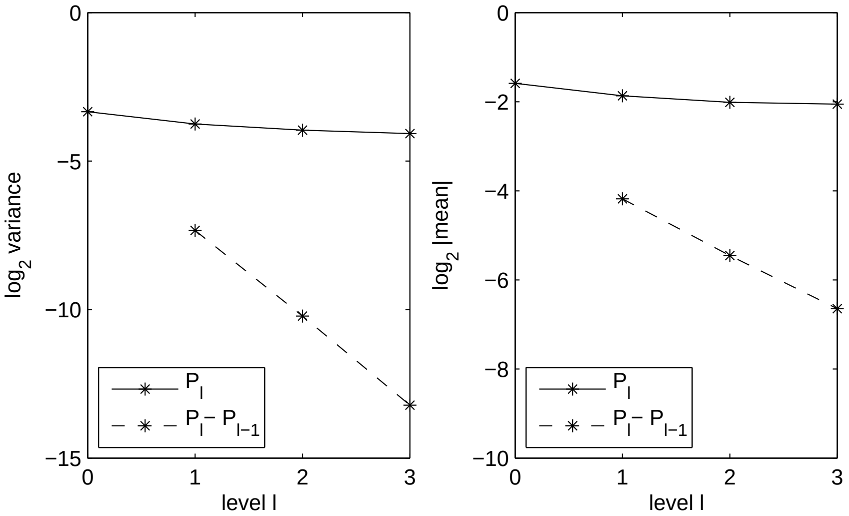
Each finer level uses four times as many timesteps, and twice as many spatial points, due to numerical stability constraints.

Parabolic SPDE

- coarsest level of approximation uses 1 timestep per quarter, and 10 spatial points
- implementation is really very easy – most interesting part of research was mean-square stability theory, with and without absorbing boundary
- computational cost $C_\ell \propto 8^\ell$
- numerical results suggest variance $V_\ell \propto 8^{-\ell}$
- can prove $V_\ell \propto 16^{-\ell}$ when no absorbing boundary

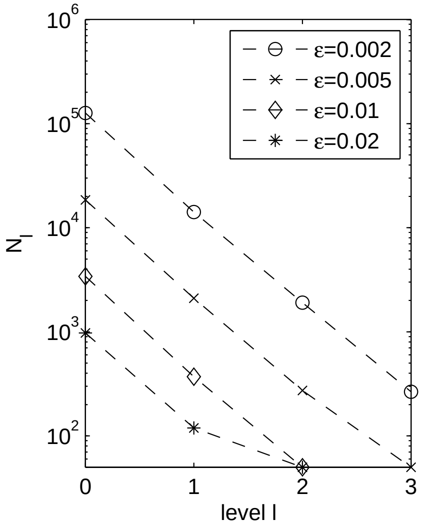
Parabolic SPDE

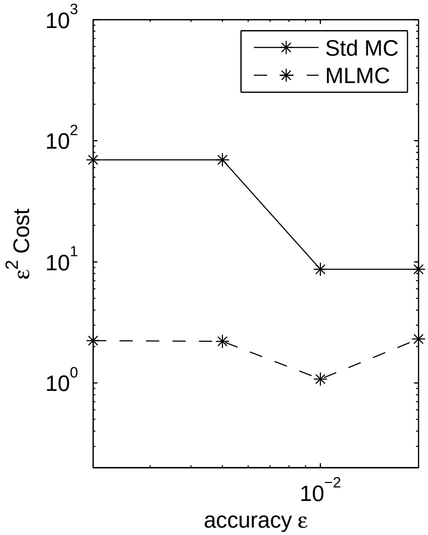
Fractional loss on equity tranche of a 5-year CDO:



Parabolic SPDE

Fractional loss on equity tranche of a 5-year CDO:





PDEs with Uncertainty

I'm working with Rob Scheichl (Bath) and Andrew Cliffe (Nottingham) on multilevel Monte Carlo for the modelling of oil reservoirs and groundwater contamination in nuclear waste repositories.

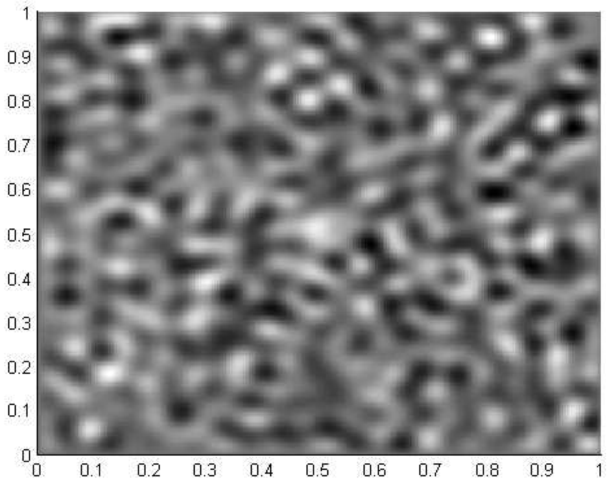
Here we have an elliptic SPDE coming from Darcy's law:

$$\nabla \cdot \Big(\kappa(x) \nabla p \Big) = 0$$

where the permeability $\kappa(x)$ is uncertain, and $\log \kappa(x)$ is often modelled as being Normally distributed with a spatial covariance such as

$$\mathrm{cov}(\log \kappa(x_1), \log \kappa(x_2)) = \sigma^2 \exp(-\|x_1 - x_2\|/\lambda)$$

Elliptic SPDE



A typical realisation of κ for $\lambda = 0.001, \ \sigma = 1$.

Elliptic SPDE

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 θ_n, f_n are eigenvalues / eigenfunctions of the correlation function:

$$\int R(\mathbf{x},\mathbf{y})\ f_n(\mathbf{y})\ \mathrm{d}\mathbf{y} = \theta_n\, f_n(\mathbf{x})$$

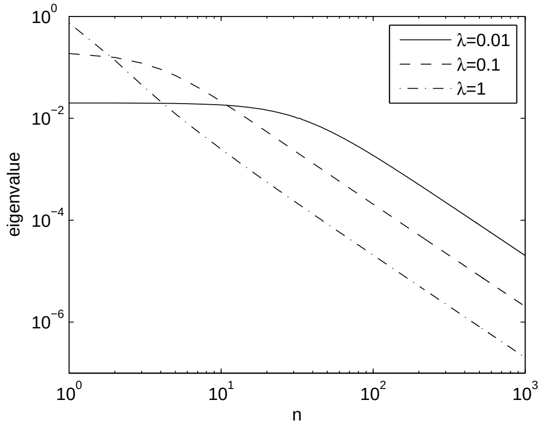
and $\xi_n(\omega)$ are standard Normal random variables.

Numerical experiments truncate the expansion.

(Latest 2D/3D work uses an efficient FFT construction based on a circulant embedding.)

Elliptic SPDE

Decay of 1D eigenvalues



When $\lambda = 1$, can use a low-dimensional polynomial chaos approach, but it's impractical for smaller λ.

Elliptic SPDE

Discretisation:

- cell-centred finite volume discretisation on a uniform grid – for rough coefficients we need to make grid spacing very small on finest grid
- each level of refinement has twice as many grid points in each direction
- current numerical experiments use a direct solver for simplicity, but in 3D will use an efficient AMG multigrid solver with a cost roughly proportional to the total number of grid points

2D Results

Boundary conditions for unit square $[0,1]^2$:
– fixed pressure: $p(0,x_2)\!=\!1,\;\; p(1,x_2)\!=\!0$
– Neumann b.c.: $\partial p/\partial x_2(x_1,0)\!=\!\partial p/\partial x_2(x_1,1)\!=\!0$

Output quantity – mass flux: $-\displaystyle\int k\,\frac{\partial p}{\partial x_1}\;\mathrm{d}x_2$

Correlation length: $\lambda = 0.2$

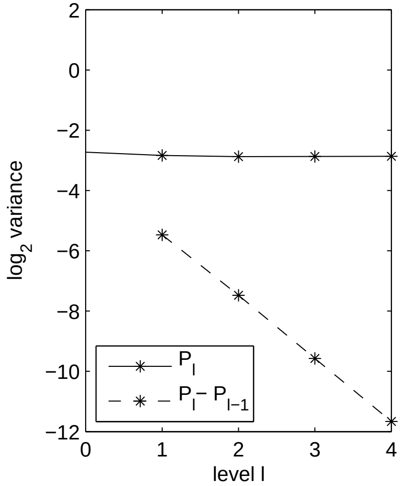
Coarsest grid: $h = 1/8$ (comparable to λ)

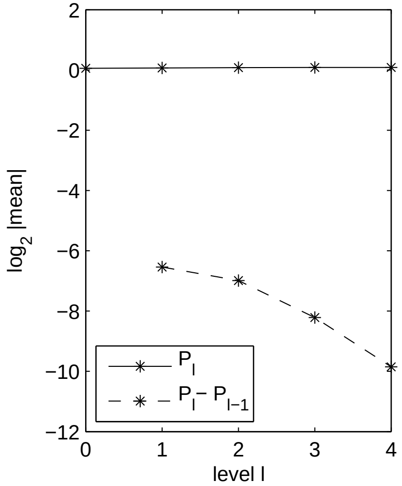
Finest grid: $h = 1/128$

Karhunen-Loève truncation: $m_{KL} = 4000$

Cost taken to be proportional to number of nodes

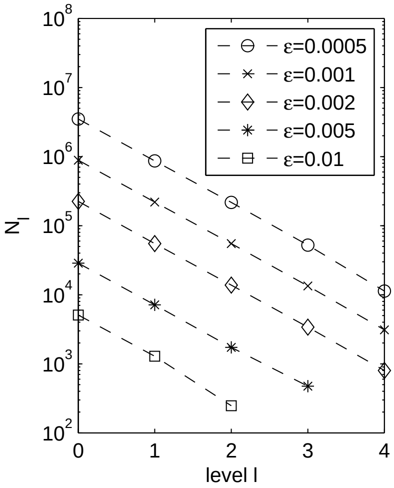
2D Results



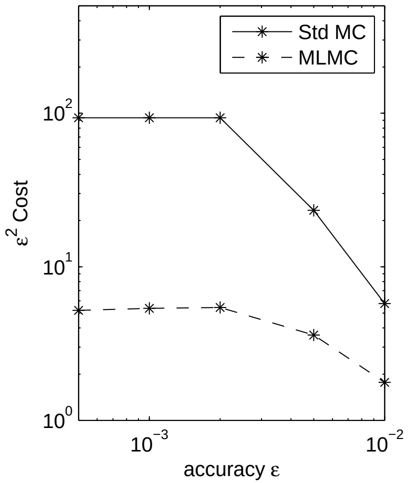


$$\mathbb{V}[\widehat{P}_\ell - \widehat{P}_{\ell-1}] \sim h_\ell^2 \qquad \mathbb{E}[\widehat{P}_\ell - \widehat{P}_{\ell-1}] \sim h_\ell^2$$

2D Results



ε=0.0005
ε=0.001
ε=0.002
ε=0.005
ε=0.01
N_l
level l



Std MC
MLMC
ε^2 Cost
accuracy ε

Complexity analysis

Relating things back to the MLMC theorem:

$$\begin{aligned}
\mathbb{E}[\widehat{P}_\ell - P] &\sim 2^{-2\ell} &\implies& \quad \alpha = 2 \\
V_\ell &\sim 2^{-2\ell} &\implies& \quad \beta = 2 \\
C_\ell &\sim 2^{d\ell} &\implies& \quad \gamma = d \qquad \text{(dimension of PDE)}
\end{aligned}$$

To achieve r.m.s. accuracy ε requires finest level grid spacing $h \sim \varepsilon^{1/2}$ and hence we get the following complexity:

dim	MC	MLMC
1	$\varepsilon^{-2.5}$	ε^{-2}
2	ε^{-3}	$\varepsilon^{-2}(\log \varepsilon)^2$
3	$\varepsilon^{-3.5}$	$\varepsilon^{-2.5}$

Other SPDE applications

For more on multilevel for SPDEs, see the work of Christoph Schwab and his group (ETH Zurich):

`http://www.math.ethz.ch/∼schwab/`

- elliptic, parabolic and hyperbolic PDEs
- stochastic coefficients, initial data, boundary data

Schwab used to work on alternative techniques such as “polynomial chaos” but has now switched to multilevel because of its superior efficiency for many applications.

For other papers on multilevel, see my MLMC community homepage:

`http://people.maths.ox.ac.uk/gilesm/mlmc_community.html`

Details of MATLAB MLMC code

In Practical 2 you will be working with a set of MATLAB codes which implement the multilevel Monte Carlo method for 4 applications:

- financial option based on Geometric Brownian Motion SDE
- 1D elliptic PDE with uncertain forcing / boundary conditions
- 1D parabolic PDE with uncertain forcing / initial data / b.c.'s
- 1D nonlinear hyperbolic PDE with uncertain initial data / b.c.'s

These all use

- `mlmc.m`: "driver" code which performs the MLMC calculation using a user routine to estimate $\mathbb{E}[P_\ell - P_{\ell-1}]$ using N_ℓ samples
- `mlmc_test.m`: a program which does a lot of tests and then calls `mlmc.m` to perform a number of MLMC calculations

Details of MATLAB MLMC code

`mlmc_test.m` first performs a set of calculations using a fixed number of samples on each level of resolution, and produces 4 plots:

- $\log_2(V_\ell)$ versus level ℓ
- $\log_2(|\mathbb{E}[P_\ell - P_{\ell-1}]|)$ versus level ℓ
- consistency check versus level
- kurtosis versus level

Details of MATLAB MLMC code

If a, b, c are estimates for $\mathbb{E}[P_{\ell-1}]$, $\mathbb{E}[P_\ell]$, $\mathbb{E}[P_\ell - P_{\ell-1}]$, respectively, then it should be true that $a - b + c \approx 0$

The consistency check verifies that this is true, to within the accuracy one would expect due to sampling error.

Since

$$\sqrt{\mathbb{V}[a-b+c]} \leq \sqrt{\mathbb{V}[a]} + \sqrt{\mathbb{V}[b]} + \sqrt{\mathbb{V}[c]}$$

it computes the ratio

$$\frac{|a-b+c|}{3\left(\sqrt{\mathbb{V}[a]} + \sqrt{\mathbb{V}[b]} + \sqrt{\mathbb{V}[c]}\right)}$$

The probability of this ratio being greater than 1 based on random sampling errors is extremely small. If it is, it indicates a likely error.

Details of MATLAB MLMC code

The MLMC approach needs a good estimate for $V_\ell = \mathbb{V}[P_\ell - P_{\ell-1}]$, but how many samples are need for this?

As few as 10 may be sufficient in many cases for a rough estimate, but many more are needed when there are rare outliers.

When the number of samples N is large, the standard deviation of the sample variance for a random variable X with zero mean is approximately

$$\sqrt{\frac{\kappa-1}{N}}\,\mathbb{E}[X^2] \quad \text{where kurtosis } \kappa \text{ is defined as} \quad \kappa = \frac{\mathbb{E}[X^4]}{(\mathbb{E}[X^2])^2}$$

(see `http://mathworld.wolfram.com/SampleVarianceDistribution.html`)

As well as plotting κ_ℓ, `mlmc_test.m` will give a warning if κ_ℓ is very large.

Details of MATLAB MLMC code

An extreme (but important) example is a digital option in which P always takes the value 0 or 1.

In this case we have

$$X \equiv P_\ell - P_{\ell-1} = \begin{cases} 1, & \text{probability } p \\ -1, & \text{probability } q \\ 0, & \text{probability } 1-p-q \end{cases}$$

If $p, q \ll 1$, then $\mathbb{E}[X] \approx 0$, and

$$\kappa \approx \frac{p+q}{(p+q)^2} = (p+q)^{-1} \gg 1$$

Therefore, many samples are required for a good estimate of V_ℓ, and if we don't have many samples, we may get all $X^{(n)} = 0$, which will give an estimated variance of zero.

Details of MATLAB MLMC code

```
% function [P, Nl] = mlmc(N0,eps,mlmc_l, alpha,beta,gamma)
%
% P       = value
% Nl      = number of samples at each level
% N0      = initial number of samples on levels 0,1,2
% eps     = desired accuracy (rms error)
% alpha   -> weak error is  O(2^{-alpha*l})
% beta    -> variance is    O(2^{-beta*l})
% gamma   -> sample cost is O(2^{gamma*l})
%
% if alpha, beta are not positive then they will be estimated
%
% mlmc_l = function for level l estimator
%
% sums = mlmc_fn(l,N)     low-level routine
% inputs:  l = level
%          N = number of paths
% output: sums(1) = sum(Y)
%         sums(2) = sum(Y.^2)
%         where Y are iid samples with expected value:
%         E[P_0] on level 0
%         E[P_l - P_{l-1}] on level l>0
```

Details of MATLAB MLMC code

```
function [P, Nl] = mlmc(N0,eps,mlmc_l, alpha_0,beta_0,gamma)

  alpha = max(0, alpha_0);
  beta  = max(0, beta_0);

  L             = 2;
  Nl(1:3)       = 0;
  suml(1:2,1:3) = 0;
  dNl(1:3)      = N0;

  while sum(dNl) > 0

%
% update sample sums
%
    for l=0:L
      if dNl(l+1) > 0
        sums        = feval(mlmc_l,l,dNl(l+1));
        Nl(l+1)     = Nl(l+1) + dNl(l+1);
        suml(1,l+1) = suml(1,l+1) + sums(1);
        suml(2,l+1) = suml(2,l+1) + sums(2);
      end
    end
```

Details of MATLAB MLMC code

```
%
% compute absolute average and variance
%
    ml = abs(   suml(1,:)./Nl);
    Vl = max(0, suml(2,:)./Nl - ml.^2);

%
% fix to cope with possible zero values for ml and Vl
% (can happen in some applications when there are few samples)
%
    for l = 3:L+1
      ml(l) = max(ml(l), 0.5*ml(l-1)/2^alpha);
      Vl(l) = max(Vl(l), 0.5*Vl(l-1)/2^beta);
    end
```

Details of MATLAB MLMC code

```
%
% use linear regression to estimate alpha, beta if not given
%
    if alpha_0 <= 0
      A     = repmat((1:L)',1,2).^repmat(1:-1:0,L,1);
      x     = A \ log2(ml(2:end))';
      alpha = max(0.5,-x(1));
    end

    if beta_0 <= 0
      A    = repmat((1:L)',1,2).^repmat(1:-1:0,L,1);
      x    = A \ log2(Vl(2:end))';
      beta = max(0.5,-x(1));
    end
```

Details of MATLAB MLMC code

```
%
% set optimal number of additional samples
%
    Cl  = 2.^(gamma*(0:L));
    Ns  = ceil(2 * sqrt(Vl./Cl) * sum(sqrt(Vl.*Cl)) / eps^2);
    dNl = max(0, Ns-Nl);
%
```

Here C_ℓ is the cost per sample. The optimal number of samples N_ℓ is chosen to minimise $\displaystyle\sum_\ell N_\ell C_\ell$ subject to $\displaystyle\sum_\ell N_\ell^{-1} V_\ell \approx \tfrac{1}{2}\varepsilon^2$.

Using a Lagrange multiplier gives $N_\ell = \lambda\sqrt{V_\ell/C_\ell}$ where

$$\lambda^{-1}\sum_\ell \sqrt{V_\ell\, C_\ell} = \tfrac{1}{2}\varepsilon^2.$$

Details of MATLAB MLMC code

```
% if (almost) converged, estimate remaining error and decide
% whether a new level is required
%
    if sum( dNl > 0.01*Nl ) == 0
      range = -2:0;
      rem = max(ml(L+1+range).*2.^(alpha*range)) / (2^alpha - 1);

      if rem > eps/sqrt(2)
        L       = L+1;
        Vl(L+1) = Vl(L) / 2^beta;
        Nl(L+1) = 0;
        suml(1:4,L+1) = 0;

        Cl  = 2.^(gamma*(0:L));
        Ns  = ceil(2 * sqrt(Vl./Cl) * sum(sqrt(Vl.*Cl)) / eps^2);
        dNl = max(0, Ns-Nl);
      end
    end
  end
%
% finally, evaluate multilevel estimator
%
  P = sum(suml(1,:)./Nl);
```

Details of MATLAB MLMC code

If $\mathbb{E}[P_\ell - P_{\ell-1}] \propto 2^{-\alpha\ell}$ then the remaining error is

$$\mathbb{E}[P - P_L] = \sum_{\ell=L+1}^{\infty} \mathbb{E}[P_\ell - P_{\ell-1}] \approx \mathbb{E}[P_L - P_{L-1}] \sum_{k=1}^{\infty} 2^{-\alpha k}$$
$$= \mathbb{E}[P_L - P_{L-1}] \, / \, (2^\alpha - 1)$$

We want $|\mathbb{E}[P - P_L]| < \varepsilon/\sqrt{2}$ (remember the two contributions to the Mean Square Error), so that gives the convergence test

$$|\mathbb{E}[P_L - P_{L-1}]| \; / \; (2^\alpha - 1) < \varepsilon/\sqrt{2}$$

For robustness, we extend this check to extrapolate from the previous two data points $\mathbb{E}[P_{L-1} - P_{L-2}]$, $\mathbb{E}[P_{L-2} - P_{L-3}]$, and take the maximum over all three as the estimated remaining error.

Extension to multiple outputs

The analysis so far has considered a single output quantity

What do we do if we want more than one?

The main approach stays exactly the same – the main change is in how we choose the number of samples on each level of approximation

Extension to multiple outputs

Suppose we have M outputs and want

$$\sum_{\ell=0}^{L} N_\ell^{-1} V_{\ell,m} \;\leq\; \tfrac{1}{2}\,\varepsilon_m^2, \qquad m = 1, \ldots, M,$$

where $V_{\ell,m}$ is the variance of the multilevel correction for output m, and ε_m is the desired RMS accuracy for that output.

As usual the computational cost is

$$\sum_{\ell=0}^{L} N_\ell\, C_\ell,$$

and we can then do a constrained optimisation using M Lagrange multipliers.

However, this is a bit nasty – it's not clear how many of the Lagrange multipliers will be "active"

Extension to multiple outputs

A simpler approach is to define

$$V_\ell = \max_m \frac{V_{\ell,m}}{\varepsilon_m^2}$$

and make the variance constraint

$$\sum_{\ell=0}^{L} N_\ell^{-1} \, V_\ell \;\leq\; \tfrac{1}{2}.$$

This brings it back to a problem with a single Lagrange multiplier with the same optimal solution as before.

Klaus Ritter and Tigran Nagapetyan (Kaiserslautern) are using this to estimate the CDF (cumulative distribution function) of an exit time τ. We estimate the CDF at a set of exit times τ_k, and then use a cubic spline to approximate the full CDF.

Non-geometric multilevel

Almost all applications of multilevel in the literature so far use a geometric sequence of levels, refining the timestep (or the spatial discretisation for PDEs) by a constant factor when going from level ℓ to level $\ell+1$.

Coming from a multigrid background, this is very natural, but it is **NOT** a requirement of the multilevel Monte Carlo approach.

All MLMC needs is a sequence of levels with

- increasing accuracy
- increasing cost
- increasingly small difference between outputs on successive levels

Non-geometric multilevel

At MIT, Nyugen is starting to look at an application based on reduced order modelling in which a solution can be expressed as a sum over a number of modes:

$$u = \sum_{m=1}^{M} a_m \; u_m$$

Here the u_m are fixed, and the amplitudes a_m are calculated by solving a reduced order problem which depends on some stochastic inputs.

Increasing M increases the accuracy, but also increases the cost.

MLMC may be very effective, but it's not at all clear how to choose the levels. Geometric might be OK:

$$M_\ell = \{1,\ 2,\ 4,\ 8,\ 16\}$$

but perhaps linear would be better?

$$M_\ell = \{2,\ 4,\ 6,\ 8,\ 10,\ 12\}$$