Multilevel Monte Carlo methods for Uncertainty Quantification

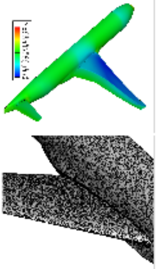
Fabio Nobile

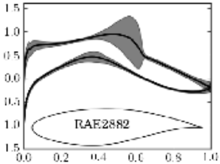
CSQI - Institute of Mathematics, EPFL, Switzerland

CEA/EDF/INRIA Summer School
Multi-fidélité, multi-niveaux, sélection / agrégation de modèles
Université Sorbonne – Jussieu, June 14-18, 2021

Outline

UQ analysis for complex models





Working assumptions:

- Complex computational models:a single scenario analysis is computationally heavy
- multiple scenarios investigation and UQ analysis is often unaffordable
- **Question**: how to exploit multiple discretizations with different accuracy levels to reduce the cost of UQ analysis

EPFL

Problem setting – forward uncertainty propagation

- Random input parameters: $y \in \Gamma$ with given distribution
  Assumption 1: we can sample y exactly and independently.
- Complex differential model (e.g. Euler, Navier-Stokes, elastodynamics, ...):

$$\mathcal{L}_y u = \mathcal{F}_y \tag{1}$$

  Assumption 2: for any $y \in \Gamma$, (1) has a unique solution $u = u(y) \in V$ (V: solution space, typically a Banach space)
- (random) Output quantity of interest (e.g. lift, drag, etc.):

$$Q(y) = \tilde{Q}(y, u(y)) \in \mathbb{R}, \quad \forall y \in \Gamma$$

Goal: compute $\mu = \mathbb{E}[Q] = \mathbb{E}_y[\tilde{Q}(y, u(y))]$ or other statistical quantities

In practice, u is not accessible and can only be computed approximately.

Computational model

$$\mathcal{L}_{h,y} u_h = \mathcal{F}_{y,h}, \qquad \text{Computational output} \quad Q_h(y) = \tilde{Q}(y, u_h(y))$$

h: discretization parameter (e.g. mesh size); $Q_h(y) \xrightarrow{h \to 0} Q(y)$, $\forall y \in \Gamma$

Problem setting – forward uncertainty propagation

- Random input parameters: $y \in \Gamma$ with given distribution
  Assumption 1: we can sample y exactly and independently.
- Complex differential model (e.g. Euler, Navier-Stokes, elastodynamics, ...):

$$\mathcal{L}_y u = \mathcal{F}_y \tag{1}$$

  Assumption 2: for any $y \in \Gamma$, (1) has a unique solution $u = u(y) \in V$ (V: solution space, typically a Banach space)
- (random) Output quantity of interest (e.g. lift, drag, etc.):

$$Q(y) = \tilde{Q}(y, u(y)) \in \mathbb{R}, \quad \forall y \in \Gamma$$

Goal: compute $\mu = \mathbb{E}[Q] = \mathbb{E}_y[\tilde{Q}(y, u(y))]$ or other statistical quantities

In practice, u is not accessible and can only be computed approximately.

Computational model

$$\mathcal{L}_{h,y} u_h = \mathcal{F}_{y,h}, \qquad \text{Computational output} \quad Q_h(y) = \tilde{Q}(y, u_h(y))$$

h: discretization parameter (e.g. mesh size); $Q_h(y) \xrightarrow{h \to 0} Q(y)$, $\forall y \in \Gamma$

Problem setting – forward uncertainty propagation

- Random input parameters: $y \in \Gamma$ with given distribution
  Assumption 1: we can sample y exactly and independently.
- Complex differential model (e.g. Euler, Navier-Stokes, elastodynamics, ...):

$$\mathcal{L}_y u = \mathcal{F}_y \tag{1}$$

  Assumption 2: for any $y \in \Gamma$, (1) has a unique solution $u = u(y) \in V$
  (V: solution space, typically a Banach space)
- (random) Output quantity of interest (e.g. lift, drag, etc.):

$$Q(y) = \tilde{Q}(y, u(y)) \in \mathbb{R}, \quad \forall y \in \Gamma$$

Goal: compute $\mu = \mathbb{E}[Q] = \mathbb{E}_y[\tilde{Q}(y, u(y))]$ or other statistical quantities

In practice, u is not accessible and can only be computed approximately.

Computational model

$$\mathcal{L}_{h,y} u_h = \mathcal{F}_{y,h}, \qquad \text{Computational output} \quad Q_h(y) = \tilde{Q}(y, u_h(y))$$

h: discretization parameter (e.g. mesh size); $Q_h(y) \xrightarrow{h \to 0} Q(y)$, $\forall y \in \Gamma$

Problem setting – forward uncertainty propagation

- Random input parameters: $y \in \Gamma$ with given distribution
  Assumption 1: we can sample y exactly and independently.
- Complex differential model (e.g. Euler, Navier-Stokes, elastodynamics, ...):

$$\mathcal{L}_y u = \mathcal{F}_y \tag{1}$$

  Assumption 2: for any $y \in \Gamma$, (1) has a unique solution $u = u(y) \in V$ (V: solution space, typically a Banach space)
- (random) Output quantity of interest (e.g. lift, drag, etc.):

$$Q(y) = \tilde{Q}(y, u(y)) \in \mathbb{R}, \quad \forall y \in \Gamma$$

Goal: compute $\mu = \mathbb{E}[Q] = \mathbb{E}_y[\tilde{Q}(y, u(y))]$ or other statistical quantities

In practice, u is not accessible and can only be computed approximately.

Computational model

$$\mathcal{L}_{h,y} u_h = \mathcal{F}_{y,h}, \qquad \text{Computational output} \quad Q_h(y) = \tilde{Q}(y, u_h(y))$$

h: discretization parameter (e.g. mesh size); $Q_h(y) \xrightarrow{h \to 0} Q(y)$, $\forall y \in \Gamma$

EPFL

Problem setting – forward uncertainty propagation

- Random input parameters: $y \in \Gamma$ with given distribution
  Assumption 1: we can sample y exactly and independently.
- Complex differential model (e.g. Euler, Navier-Stokes, elastodynamics, ...):

$$\mathcal{L}_y u = \mathcal{F}_y \tag{1}$$

  Assumption 2: for any $y \in \Gamma$, (1) has a unique solution $u = u(y) \in V$
  (V: solution space, typically a Banach space)
- (random) Output quantity of interest (e.g. lift, drag, etc.):

$$Q(y) = \tilde{Q}(y, u(y)) \in \mathbb{R}, \quad \forall y \in \Gamma$$

Goal: compute $\mu = \mathbb{E}[Q] = \mathbb{E}_y[\tilde{Q}(y, u(y))]$ or other statistical quantities

In practice, u is not accessible and can only be computed approximately.

Computational model

$$\mathcal{L}_{h,y} u_h = \mathcal{F}_{y,h}, \qquad \text{Computational output} \quad Q_h(y) = \tilde{Q}(y, u_h(y))$$

h: discretization parameter (e.g. mesh size); $Q_h(y) \xrightarrow{h \to 0} Q(y)$, $\forall y \in \Gamma$

Monte Carlo method

- Generate M iid copies $y^{(1)}, \dots, y^{(M)} \sim y$
- Compute the corresponding outputs $Q_h(y^{(i)}), \;\; i = 1, \dots, M$
- Approximate expectation by sample average $\hat{\mathbb{E}}_M[\cdot]$

$$\hat{\mu}_{MC} := \hat{\mathbb{E}}_M[Q_h] = \frac{1}{M} \sum_{i=1}^{M} Q_h^{(i)}$$

Bias (discretization error):

$$B := \mathbb{E}[\hat{\mu}_{MC}] - \mu = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[Q_h(y^{(i)})] - \mathbb{E}[Q] = \mathbb{E}[Q_h] - \mathbb{E}[Q]$$

The estimator is biased, in general, because of the discretization error

Variance (statistical error):

$$V_{MC} := \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}])^2] = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[(Q_h(y^{(i)}) - \mathbb{E}[Q_h])^2] = \frac{\mathbb{V}\mathrm{ar}[Q_h]}{M}$$

Typical Monte Carlo variance decay $O(\frac{1}{M})$

Monte Carlo method

- Generate M iid copies $y^{(1)}, \dots, y^{(M)} \sim y$
- Compute the corresponding outputs $Q_h(y^{(i)}), \;\; i = 1, \dots, M$
- Approximate expectation by sample average $\hat{\mathbb{E}}_M[\cdot]$

$$\hat{\mu}_{MC} := \hat{\mathbb{E}}_M[Q_h] = \frac{1}{M} \sum_{i=1}^{M} Q_h^{(i)}$$

Bias (discretization error):

$$B := \mathbb{E}[\hat{\mu}_{MC}] - \mu = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[Q_h(y^{(i)})] - \mathbb{E}[Q] = \mathbb{E}[Q_h] - \mathbb{E}[Q]$$

The estimator is biased, in general, because of the discretization error

Variance (statistical error):

$$V_{MC} := \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}])^2] = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[(Q_h(y^{(i)}) - \mathbb{E}[Q_h])^2] = \frac{\mathrm{Var}[Q_h]}{M}$$

Typical Monte Carlo variance decay $O(\frac{1}{M})$

Monte Carlo method

- Generate M iid copies $y^{(1)}, \dots, y^{(M)} \sim y$
- Compute the corresponding outputs $Q_h(y^{(i)}), \;\; i = 1, \dots, M$
- Approximate expectation by sample average $\hat{\mathbb{E}}_M[\cdot]$

$$\hat{\mu}_{MC} := \hat{\mathbb{E}}_M[Q_h] = \frac{1}{M} \sum_{i=1}^{M} Q_h^{(i)}$$

Bias (discretization error):

$$B := \mathbb{E}[\hat{\mu}_{MC}] - \mu = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[Q_h(y^{(i)})] - \mathbb{E}[Q] = \mathbb{E}[Q_h] - \mathbb{E}[Q]$$

The estimator is biased, in general, because of the discretization error

Variance (statistical error):

$$V_{MC} := \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}])^2] = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[(Q_h(y^{(i)}) - \mathbb{E}[Q_h])^2] = \frac{\mathbb{V}ar[Q_h]}{M}$$

Typical Monte Carlo variance decay $O(\frac{1}{M})$

Monte Carlo method

- Generate M iid copies $y^{(1)}, \dots, y^{(M)} \sim y$
- Compute the corresponding outputs $Q_h(y^{(i)}),\ \ i = 1, \dots, M$
- Approximate expectation by sample average $\hat{\mathbb{E}}_M[\cdot]$

$$\hat{\mu}_{MC} := \hat{\mathbb{E}}_M[Q_h] = \frac{1}{M}\sum_{i=1}^{M} Q_h^{(i)}$$

Bias (discretization error):

$$B := \mathbb{E}[\hat{\mu}_{MC}] - \mu = \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}[Q_h(y^{(i)})] - \mathbb{E}[Q] = \mathbb{E}[Q_h] - \mathbb{E}[Q]$$

The estimator is biased, in general, because of the discretization error

Variance (statistical error):

$$V_{MC} := \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}])^2] = \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}[(Q_h(y^{(i)}) - \mathbb{E}[Q_h])^2] = \frac{Var[Q_h]}{M}$$

Typical Monte Carlo variance decay $O(\frac{1}{M})$

EPFL

Monte Carlo method

- Generate M iid copies $y^{(1)}, \dots, y^{(M)} \sim y$
- Compute the corresponding outputs $Q_h(y^{(i)}),\ \ i = 1, \dots, M$
- Approximate expectation by sample average $\hat{\mathbb{E}}_M[\cdot]$

$$\hat{\mu}_{MC} := \hat{\mathbb{E}}_M[Q_h] = \frac{1}{M}\sum_{i=1}^{M} Q_h^{(i)}$$

Bias (discretization error):

$$B := \mathbb{E}[\hat{\mu}_{MC}] - \mu = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}[Q_h(y^{(i)})] - \mathbb{E}[Q] = \mathbb{E}[Q_h] - \mathbb{E}[Q]$$

The estimator is biased, in general, because of the discretization error

Variance (statistical error):

$$V_{MC} := \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}])^2] = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}[(Q_h(y^{(i)}) - \mathbb{E}[Q_h])^2] = \frac{\mathrm{Var}[Q_h]}{M}$$

Typical Monte Carlo variance decay $O(\frac{1}{M})$

Monte Carlo method

- Generate M iid copies $y^{(1)}, \dots, y^{(M)} \sim y$
- Compute the corresponding outputs $Q_h(y^{(i)}),\ \ i = 1, \dots, M$
- Approximate expectation by sample average $\hat{\mathbb{E}}_M[\cdot]$

$$\hat{\mu}_{MC} := \hat{\mathbb{E}}_M[Q_h] = \frac{1}{M}\sum_{i=1}^{M} Q_h^{(i)}$$

Bias (discretization error):

$$B := \mathbb{E}[\hat{\mu}_{MC}] - \mu = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}[Q_h(y^{(i)})] - \mathbb{E}[Q] = \mathbb{E}[Q_h] - \mathbb{E}[Q]$$

The estimator is biased, in general, because of the discretization error

Variance (statistical error):

$$V_{MC} := \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}])^2] = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}[(Q_h(y^{(i)}) - \mathbb{E}[Q_h])^2] = \frac{\mathbb{V}\mathrm{ar}[Q_h]}{M}$$

Typical Monte Carlo variance decay $O(\frac{1}{M})$

Monte Carlo method

- Generate M iid copies $y^{(1)}, \dots, y^{(M)} \sim y$
- Compute the corresponding outputs $Q_h(y^{(i)}), \;\; i = 1, \dots, M$
- Approximate expectation by sample average $\hat{\mathbb{E}}_M[\cdot]$

$$\hat{\mu}_{MC} := \hat{\mathbb{E}}_M[Q_h] = \frac{1}{M} \sum_{i=1}^{M} Q_h^{(i)}$$

Bias (discretization error):

$$B := \mathbb{E}[\hat{\mu}_{MC}] - \mu = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[Q_h(y^{(i)})] - \mathbb{E}[Q] = \mathbb{E}[Q_h] - \mathbb{E}[Q]$$

The estimator is biased, in general, because of the discretization error

Variance (statistical error):

$$V_{MC} := \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}])^2] = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[(Q_h(y^{(i)}) - \mathbb{E}[Q_h])^2] = \frac{\mathbb{V}\mathrm{ar}[Q_h]}{M}$$

Typical Monte Carlo variance decay $O(\frac{1}{M})$

Quantifying the error in Monte Carlo

Mean squared error

$$\mathrm{MSE}(\hat{\mu}_{MC}) := \mathbb{E}[(\hat{\mu}_{MC} - \mu)^2] = \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}] + \mathbb{E}[\hat{\mu}_{MC}] - \mu)^2]$$
$$= V_{MC} + B^2$$
$$= \frac{\mathbb{V}\mathrm{ar}[Q_h]}{M} + \mathbb{E}[Q_h - Q]^2$$

Controlling the MSE

- Bias estimation: needs an error estimator $\eta_h(y) \approx Q_h(y) - Q(y)$, e.g.
  - goal oriented a posteriori error estimator (dual weighted residual based)
  - $Q_h(y) - Q^*(y)$ with $Q^*(y)$ a Richardson extrapolation from $Q_h(y), Q_{2h}(y)$

  Then, $$B \approx \hat{B} := \hat{\mathbb{E}}_M[\eta_h] = \frac{1}{M}\sum_{i=1}^{M} \eta_h(y^{(i)})$$

- Variance estimation: one can use the sample variance estimator $\hat{V}_h := \widehat{\mathbb{V}\mathrm{ar}}_M[Q_h] = \frac{1}{M-1}\sum_{i=1}^{M}(Q_h(y^{(i)}) - \hat{\mu}_{MC})^2$. Then

$$V_{MC} \approx \hat{V}_{MC} := \frac{\hat{V}_h}{M}$$

Quantifying the error in Monte Carlo

Mean squared error

$$\mathrm{MSE}(\hat{\mu}_{MC}) := \mathbb{E}[(\hat{\mu}_{MC} - \mu)^2] = \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}] + \mathbb{E}[\hat{\mu}_{MC}] - \mu)^2]$$
$$= V_{MC} + B^2$$
$$= \frac{\mathbb{V}\mathrm{ar}[Q_h]}{M} + \mathbb{E}[Q_h - Q]^2$$

Controlling the MSE

- Bias estimation: needs an error estimator $\eta_h(y) \approx Q_h(y) - Q(y)$, e.g.
  - goal oriented a posteriori error estimator (dual weighted residual based)
  - $Q_h(y) - Q^*(y)$ with $Q^*(y)$ a Richardson extrapolation from $Q_h(y), Q_{2h}(y)$

$$\text{Then,} \qquad B \approx \hat{B} := \hat{\mathbb{E}}_M[\eta_h] = \frac{1}{M}\sum_{i=1}^{M} \eta_h(y^{(i)})$$

- Variance estimation: one can use the sample variance estimator $\hat{V}_h := \hat{\mathbb{V}\mathrm{ar}}_M[Q_h] = \frac{1}{M-1}\sum_{i=1}^{M}(Q_h(y^{(i)}) - \hat{\mu}_{MC})^2$. Then

$$V_{MC} \approx \hat{V}_{MC} := \frac{\hat{V}_h}{M}$$

Quantifying the error in Monte Carlo

Mean squared error

$$\begin{aligned}\mathrm{MSE}(\hat{\mu}_{MC}) :=& \mathbb{E}[(\hat{\mu}_{MC} - \mu)^2] = \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}] + \mathbb{E}[\hat{\mu}_{MC}] - \mu)^2] \\ =& V_{MC} + B^2 \\ =& \frac{\mathbb{V}\mathrm{ar}[Q_h]}{M} + \mathbb{E}[Q_h - Q]^2\end{aligned}$$

Controlling the MSE

- Bias estimation: needs an error estimator $\eta_h(y) \approx Q_h(y) - Q(y)$, e.g.
  - goal oriented a posteriori error estimator (dual weighted residual based)
  - $Q_h(y) - Q^*(y)$ with $Q^*(y)$ a Richardson extrapolation from $Q_h(y), Q_{2h}(y)$

$$\text{Then,} \qquad B \approx \hat{B} := \hat{\mathbb{E}}_M[\eta_h] = \frac{1}{M}\sum_{i=1}^{M} \eta_h(y^{(i)})$$

- Variance estimation: one can use the sample variance estimator $\hat{V}_h := \widehat{\mathbb{V}\mathrm{ar}}_M[Q_h] = \frac{1}{M-1}\sum_{i=1}^{M}(Q_h(y^{(i)}) - \hat{\mu}_{MC})^2$. Then

$$V_{MC} \approx \hat{V}_{MC} := \frac{\hat{V}_h}{M}$$

Quantifying the error in Monte Carlo

Mean squared error

$$\begin{aligned}\mathrm{MSE}(\hat{\mu}_{MC}) :=&\mathbb{E}[(\hat{\mu}_{MC}-\mu)^2] = \mathbb{E}[(\hat{\mu}_{MC}-\mathbb{E}[\hat{\mu}_{MC}]+\mathbb{E}[\hat{\mu}_{MC}]-\mu)^2]\\ =&V_{MC}+B^2\\ =&\frac{\mathbb{V}\mathrm{ar}[Q_h]}{M}+\mathbb{E}[Q_h-Q]^2\end{aligned}$$

Controlling the MSE

- Bias estimation: needs an error estimator $\eta_h(y) \approx Q_h(y) - Q(y)$, e.g.
  - goal oriented a posteriori error estimator (dual weighted residual based)
  - $Q_h(y) - Q^*(y)$ with $Q^*(y)$ a Richardson extrapolation from $Q_h(y)$, $Q_{2h}(y)$

  Then, $$B \approx \hat{B} := \hat{\mathbb{E}}_M[\eta_h] = \frac{1}{M}\sum_{i=1}^{M} \eta_h(y^{(i)})$$
- Variance estimation: one can use the sample variance estimator $\hat{V}_h := \hat{\mathbb{V}\mathrm{ar}}_M[Q_h] = \frac{1}{M-1}\sum_{i=1}^{M}(Q_h(y^{(i)}) - \hat{\mu}_{MC})^2$. Then

$$V_{MC} \approx \hat{V}_{MC} := \frac{\hat{V}_h}{M}$$

Quantifying the error in Monte Carlo

Mean squared error

$$\begin{aligned}\mathrm{MSE}(\hat{\mu}_{MC}) :=& \mathbb{E}[(\hat{\mu}_{MC}-\mu)^2] = \mathbb{E}[(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}] + \mathbb{E}[\hat{\mu}_{MC}] - \mu)^2] \\ =& V_{MC} + B^2 \\ =& \frac{\mathbb{V}\mathrm{ar}[Q_h]}{M} + \mathbb{E}[Q_h - Q]^2\end{aligned}$$

Controlling the MSE

- Bias estimation: needs an error estimator $\eta_h(y) \approx Q_h(y) - Q(y)$, e.g.
  - goal oriented a posteriori error estimator (dual weighted residual based)
  - $Q_h(y) - Q^*(y)$ with $Q^*(y)$ a Richardson extrapolation from $Q_h(y), Q_{2h}(y)$

  Then, $$B \approx \hat{B} := \hat{\mathbb{E}}_M[\eta_h] = \frac{1}{M}\sum_{i=1}^{M} \eta_h(y^{(i)})$$
- Variance estimation: one can use the sample variance estimator $\hat{V}_h := \hat{\mathbb{V}\mathrm{ar}}_M[Q_h] = \frac{1}{M-1}\sum_{i=1}^{M}(Q_h(y^{(i)}) - \hat{\mu}_{MC})^2$. Then

  $$V_{MC} \approx \hat{V}_{MC} := \frac{\hat{V}_h}{M}$$

Quantifying the error in Monte Carlo

Using the **Central Limit Theorem (CLT)**

$$\sqrt{M}(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}]) \xrightarrow{d} N(0, V_h)$$

Asymprotic confidence interval: with probability at least $1 - \delta$

$$|\hat{\mu}_{MC} - \mu| \leq |\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}]| + |\mathbb{E}[\hat{\mu}_{MC}] - \mu|$$

$$\sim c_\delta \frac{\sqrt{V_h}}{\sqrt{M}} + |B| \approx c_\delta \frac{\sqrt{\hat{V}_h}}{\sqrt{M}} + |\hat{B}|$$

with c_δ the $(1 - \frac{\delta}{2})$-quantile of the normal distribution ($\phi(c_\delta) = 1 - \frac{\delta}{2}$)

Quantifying the error in Monte Carlo

Using the **Central Limit Theorem (CLT)**

$$\sqrt{M}(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}]) \xrightarrow{d} N(0, V_h)$$

Asymprotic confidence interval: with probability at least $1-\delta$

$$|\hat{\mu}_{MC} - \mu| \leq |\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}]| + |\mathbb{E}[\hat{\mu}_{MC}] - \mu|$$

$$\sim c_\delta \frac{\sqrt{V_h}}{\sqrt{M}} + |B| \approx c_\delta \frac{\sqrt{\hat{V}_h}}{\sqrt{M}} + |\hat{B}|$$

with c_δ the $(1-\frac{\delta}{2})$-quantile of the normal distribution ($\phi(c_\delta) = 1-\frac{\delta}{2}$)

Quantifying the error in Monte Carlo

Using the **Central Limit Theorem (CLT)**

$$\sqrt{M}(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}]) \xrightarrow{d} N(0, V_h)$$

Asymprotic confidence interval: with probability at least $1 - \delta$

$$|\hat{\mu}_{MC} - \mu| \leq |\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}]| + |\mathbb{E}[\hat{\mu}_{MC}] - \mu|$$
$$\sim c_\delta \frac{\sqrt{V_h}}{\sqrt{M}} + |B| \approx c_\delta \frac{\sqrt{\hat{V}_h}}{\sqrt{M}} + |\hat{B}|$$

with c_δ the $(1 - \frac{\delta}{2})$-quantile of the normal distribution $(\phi(c_\delta) = 1 - \frac{\delta}{2})$

Quantifying the error in Monte Carlo

Using the **Central Limit Theorem (CLT)**

$$\sqrt{M}(\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}]) \xrightarrow{d} N(0, V_h)$$

Asymprotic confidence interval: with probability at least $1-\delta$

$$\begin{aligned} |\hat{\mu}_{MC} - \mu| &\leq |\hat{\mu}_{MC} - \mathbb{E}[\hat{\mu}_{MC}]| + |\mathbb{E}[\hat{\mu}_{MC}] - \mu| \\ &\sim c_\delta \frac{\sqrt{V_h}}{\sqrt{M}} + |B| \approx c_\delta \frac{\sqrt{\hat{V}_h}}{\sqrt{M}} + |\hat{B}| \end{aligned}$$

with c_δ the $(1-\frac{\delta}{2})$-quantile of the normal distribution ($\phi(c_\delta) = 1 - \frac{\delta}{2}$)

A possible adaptive Monte Carlo algorithm

1. preliminary grid convergence study: find suitable h for which $\hat{B} \leq \frac{\mathrm{tol}}{\sqrt{2}}$
2. Pilot run: compute $\hat{\mu}_{MC}^{(0)} = \frac{1}{M^{(0)}} \sum_{i=1}^{M^{(0)}} Q_h(y^{(i)})$ and estimate $\hat{V}_h^{(1)} = \frac{1}{M^{(0)}-1} \sum_{i=1}^{M^{(1)}} (Q_h(y^{(i)}) - \hat{\mu}_{MC}^{(0)})^2$. Set $k = 1$
3. while $\frac{\hat{V}_h^{(k-1)}}{M^{(k-1)}} > \frac{\mathrm{tol}^2}{2}$ do
   1. set $M^{(k)} = \mathrm{ceil}\left(\frac{2\hat{\mathrm{V}}_{\mathrm{MC}}^{(k-1)}}{\mathrm{tol}^2}\right)$
   2. compute $Q_h(y^{(i)})$, $\quad i = M^{(k-1)}+1, \ldots, M^{(k)}$
   3. update estimates of $\hat{\mu}_{MC}^{(k)}$ and $\hat{V}_h^{(k)}$ using the newly generated samples
4. output $\hat{\mu}_{MC}^{(k)}$ and $\widehat{MSE}(\hat{\mu}_{MC}^{(k)}) = \hat{B}^2 + \frac{\hat{V}_h^{(k)}}{M^{(k)}}$

A possible adaptive Monte Carlo algorithm

1. preliminary grid convergence study: find suitable h for which $\hat{B} \leq \frac{\mathrm{tol}}{\sqrt{2}}$
2. Pilot run: compute $\hat{\mu}_{MC}^{(0)} = \frac{1}{M^{(0)}} \sum_{i=1}^{M^{(0)}} Q_h(y^{(i)})$ and estimate $\hat{V}_h^{(1)} = \frac{1}{M^{(0)}-1} \sum_{i=1}^{M^{(1)}} (Q_h(y^{(i)}) - \hat{\mu}_{MC}^{(0)})^2$. Set $k = 1$
3. while $\frac{\hat{V}_h^{(k-1)}}{M^{(k-1)}} > \frac{\mathrm{tol}^2}{2}$ do
    1. set $M^{(k)} = \mathrm{ceil}\left(\frac{2\hat{V}_{\mathrm{MC}}^{(k-1)}}{\mathrm{tol}^2}\right)$
    2. compute $Q_h(y^{(i)}), \quad i = M^{(k-1)} + 1, \dots, M^{(k)}$
    3. update estimates of $\hat{\mu}_{MC}^{(k)}$ and $\hat{V}_h^{(k)}$ using the newly generated samples
4. output $\hat{\mu}_{MC}^{(k)}$ and $\widehat{MSE}(\hat{\mu}_{MC}^{(k)}) = \hat{B}^2 + \frac{\hat{V}_h^{(k)}}{M^{(k)}}$

A possible adaptive Monte Carlo algorithm

1. preliminary grid convergence study: find suitable h for which $\hat{B} \leq \frac{\mathrm{tol}}{\sqrt{2}}$
2. Pilot run: compute $\hat{\mu}_{MC}^{(0)} = \frac{1}{M^{(0)}} \sum_{i=1}^{M^{(0)}} Q_h(y^{(i)})$ and estimate $\hat{V}_h^{(1)} = \frac{1}{M^{(0)}-1} \sum_{i=1}^{M^{(1)}} (Q_h(y^{(i)}) - \hat{\mu}_{MC}^{(0)})^2$. Set $k = 1$
3. while $\frac{\hat{V}_h^{(k-1)}}{M^{(k-1)}} > \frac{\mathrm{tol}^2}{2}$ do
   1. set $M^{(k)} = \mathrm{ceil}\left(\frac{2\hat{\mathrm{V}}_{\mathrm{MC}}^{(\mathrm{k}-1)}}{\mathrm{tol}^2}\right)$
   2. compute $Q_h(y^{(i)})$, $\quad i = M^{(k-1)} + 1, \ldots, M^{(k)}$
   3. update estimates of $\hat{\mu}_{MC}^{(k)}$ and $\hat{V}_h^{(k)}$ using the newly generated samples
4. output $\hat{\mu}_{MC}^{(k)}$ and $\widehat{MSE}(\hat{\mu}_{MC}^{(k)}) = \hat{B}^2 + \frac{\hat{V}_h^{(k)}}{M^{(k)}}$

A possible adaptive Monte Carlo algorithm

1. preliminary grid convergence study: find suitable h for which $\hat{B} \leq \frac{\mathrm{tol}}{\sqrt{2}}$
2. Pilot run: compute $\hat{\mu}_{MC}^{(0)} = \frac{1}{M^{(0)}} \sum_{i=1}^{M^{(0)}} Q_h(y^{(i)})$ and estimate $\hat{V}_h^{(1)} = \frac{1}{M^{(0)}-1} \sum_{i=1}^{M^{(1)}} (Q_h(y^{(i)}) - \hat{\mu}_{MC}^{(0)})^2$. Set $k = 1$
3. while $\frac{\hat{V}_h^{(k-1)}}{M^{(k-1)}} > \frac{\mathrm{tol}^2}{2}$ do
   1. set $M^{(k)} = \mathrm{ceil}\left(\frac{2\hat{\mathrm{V}}_{\mathrm{MC}}^{(\mathrm{k}-1)}}{\mathrm{tol}^2}\right)$
   2. compute $Q_h(y^{(i)})$, $\quad i = M^{(k-1)} + 1, \dots, M^{(k)}$
   3. update estimates of $\hat{\mu}_{MC}^{(k)}$ and $\hat{V}_h^{(k)}$ using the newly generated samples
4. output $\hat{\mu}_{MC}^{(k)}$ and $\widehat{MSE}(\hat{\mu}_{MC}^{(k)}) = \hat{B}^2 + \frac{\hat{V}_h^{(k)}}{M^{(k)}}$

A possible adaptive Monte Carlo algorithm

1. preliminary grid convergence study: find suitable h for which $\hat{B} \leq \frac{\mathrm{tol}}{\sqrt{2}}$
2. Pilot run: compute $\hat{\mu}_{MC}^{(0)} = \frac{1}{M^{(0)}} \sum_{i=1}^{M^{(0)}} Q_h(y^{(i)})$ and estimate $\hat{V}_h^{(1)} = \frac{1}{M^{(0)}-1} \sum_{i=1}^{M^{(1)}} (Q_h(y^{(i)}) - \hat{\mu}_{MC}^{(0)})^2$. Set $k = 1$
3. while $\frac{\hat{V}_h^{(k-1)}}{M^{(k-1)}} > \frac{\mathrm{tol}^2}{2}$ do
   1. set $M^{(k)} = \mathrm{ceil}\left(\frac{2\hat{\mathrm{V}}_{\mathrm{MC}}^{(\mathrm{k}-1)}}{\mathrm{tol}^2}\right)$
   2. compute $Q_h(y^{(i)}), \quad i = M^{(k-1)} + 1, \ldots, M^{(k)}$
   3. update estimates of $\hat{\mu}_{MC}^{(k)}$ and $\hat{V}_h^{(k)}$ using the newly generated samples
4. output $\hat{\mu}_{MC}^{(k)}$ and $\widehat{MSE}(\hat{\mu}_{MC}^{(k)}) = \hat{B}^2 + \frac{\hat{V}_h^{(k)}}{M^{(k)}}$

A possible adaptive Monte Carlo algorithm

1. preliminary grid convergence study: find suitable h for which $\hat{B} \leq \frac{\mathrm{tol}}{\sqrt{2}}$
2. Pilot run: compute $\hat{\mu}_{MC}^{(0)} = \frac{1}{M^{(0)}} \sum_{i=1}^{M^{(0)}} Q_h(y^{(i)})$ and estimate $\hat{V}_h^{(1)} = \frac{1}{M^{(0)}-1} \sum_{i=1}^{M^{(1)}} (Q_h(y^{(i)}) - \hat{\mu}_{MC}^{(0)})^2$. Set $k = 1$
3. while $\frac{\hat{V}_h^{(k-1)}}{M^{(k-1)}} > \frac{\mathrm{tol}^2}{2}$ do
    1. set $M^{(k)} = \mathrm{ceil}\left(\frac{2\hat{\mathrm{V}}_{\mathrm{MC}}^{(\mathrm{k}-1)}}{\mathrm{tol}^2}\right)$
    2. compute $Q_h(y^{(i)}), \quad i = M^{(k-1)} + 1, \dots, M^{(k)}$
    3. update estimates of $\hat{\mu}_{MC}^{(k)}$ and $\hat{V}_h^{(k)}$ using the newly generated samples
4. output $\hat{\mu}_{MC}^{(k)}$ and $\widehat{MSE}(\hat{\mu}_{MC}^{(k)}) = \hat{B}^2 + \frac{\hat{V}_h^{(k)}}{M^{(k)}}$

A possible adaptive Monte Carlo algorithm

1. preliminary grid convergence study: find suitable h for which $\hat{B} \leq \frac{\mathrm{tol}}{\sqrt{2}}$
2. Pilot run: compute $\hat{\mu}_{MC}^{(0)} = \frac{1}{M^{(0)}} \sum_{i=1}^{M^{(0)}} Q_h(y^{(i)})$ and estimate $\hat{V}_h^{(1)} = \frac{1}{M^{(0)}-1} \sum_{i=1}^{M^{(1)}} (Q_h(y^{(i)}) - \hat{\mu}_{MC}^{(0)})^2$. Set $k = 1$
3. while $\frac{\hat{V}_h^{(k-1)}}{M^{(k-1)}} > \frac{\mathrm{tol}^2}{2}$ do
   1. set $M^{(k)} = \mathrm{ceil}\left(\frac{2\hat{\mathrm{V}}_{\mathrm{MC}}^{(\mathrm{k}-1)}}{\mathrm{tol}^2}\right)$
   2. compute $Q_h(y^{(i)}), \quad i = M^{(k-1)} + 1, \ldots, M^{(k)}$
   3. update estimates of $\hat{\mu}_{MC}^{(k)}$ and $\hat{V}_h^{(k)}$ using the newly generated samples
4. output $\hat{\mu}_{MC}^{(k)}$ and $\widehat{MSE}(\hat{\mu}_{MC}^{(k)}) = \hat{B}^2 + \frac{\hat{V}_h^{(k)}}{M^{(k)}}$

A possible adaptive Monte Carlo algorithm

Variants:

- The error may be split unevenly between bias and variance: $\hat{B}^2 \leq (1-\theta)\mathrm{tol}^2$, $\frac{\hat{V}_h}{M} \leq \theta \mathrm{tol}^2$.
- Instead of the MSE one can control the asymptotic confidence interval: $|\hat{B}| \leq \frac{\mathrm{tol}}{2}, \quad c_\delta \frac{\sqrt{\hat{V}_h}}{\sqrt{M}} \leq \frac{\mathrm{tol}}{2}$
- The previous algorithm may suffer from early termination if the variance estimate $\hat{V}_h$ is inaccurate and too small (which may happen if M is small).
- For a more robust version one could set at items 3.1-3.2 $M^{(k)} = \gamma M^{(k-1)}$ ($\gamma > 1$) and resample from scratch ($i = 1, \dots M^{(k)}$). This gives a more robust algorithm with only little extra cost:

$$\text{total sample size} \quad \mathcal{M}^{(k)} = \sum_{i=0}^{k} M^{(k)} = \sum_{i=0}^{k} M^{(0)}\gamma^k \leq \frac{\gamma}{\gamma-1} M^{(k)}.$$

A possible adaptive Monte Carlo algorithm

Variants:

- The error may be split unevenly between bias and variance: $\hat{B}^2 \leq (1-\theta)\mathrm{tol}^2$, $\frac{\hat{V}_h}{M} \leq \theta \mathrm{tol}^2$.
- Instead of the MSE one can control the asymptotic confidence interval: —$\hat{B}| \leq \frac{\mathrm{tol}}{2}$, $\quad c_\delta \frac{\sqrt{\hat{V}_h}}{\sqrt{M}} \leq \frac{\mathrm{tol}}{2}$
- The previous algorithm may suffer from early termination if the variance estimate $\hat{V}_h$ is inaccurate and too small (which may happen if M is small).
- For a more robust version one could set at items 3.1-3.2 $M^{(k)} = \gamma M^{(k-1)}$ $(\gamma > 1)$ and resample from scratch $(i = 1, \ldots M^{(k)})$. This gives a more robust algorithm with only little extra cost:

$$\text{total sample size} \quad \mathcal{M}^{(k)} = \sum_{i=0}^{k} M^{(k)} = \sum_{i=0}^{k} M^{(0)}\gamma^k \leq \frac{\gamma}{\gamma-1} M^{(k)}.$$

A possible adaptive Monte Carlo algorithm

Variants:

- The error may be split unevenly between bias and variance: $\hat{B}^2 \leq (1-\theta)\mathrm{tol}^2$, $\frac{\hat{V}_h}{M} \leq \theta \mathrm{tol}^2$.
- Instead of the MSE one can control the asymptotic confidence interval: $—\hat{B}| \leq \frac{\mathrm{tol}}{2}, \quad c_\delta \frac{\sqrt{\hat{V}_h}}{\sqrt{M}} \leq \frac{\mathrm{tol}}{2}$
- The previous algorithm may suffer from early termination if the variance estimate $\hat{V}_h$ is inaccurate and too small (which may happen if M is small).
- For a more robust version one could set at items 3.1-3.2 $M^{(k)} = \gamma M^{(k-1)}$ $(\gamma > 1)$ and resample from scratch $(i = 1, \ldots M^{(k)})$. This gives a more robust algorithm with only little extra cost:

$$\text{total sample size} \quad \mathcal{M}^{(k)} = \sum_{i=0}^{k} M^{(k)} = \sum_{i=0}^{k} M^{(0)}\gamma^k \leq \frac{\gamma}{\gamma-1} M^{(k)}.$$

A possible adaptive Monte Carlo algorithm

Variants:

- The error may be split unevenly between bias and variance: $\hat{B}^2 \leq (1-\theta)\mathrm{tol}^2$, $\frac{\hat{V}_h}{M} \leq \theta \mathrm{tol}^2$.
- Instead of the MSE one can control the asymptotic confidence interval: —$\hat{B}| \leq \frac{\mathrm{tol}}{2}$, $\quad c_\delta \frac{\sqrt{\hat{V}_h}}{\sqrt{M}} \leq \frac{\mathrm{tol}}{2}$
- The previous algorithm may suffer from early termination if the variance estimate $\hat{V}_h$ is inaccurate and too small (which may happen if M is small).
- For a more robust version one could set at items 3.1-3.2 $M^{(k)} = \gamma M^{(k-1)}$ $(\gamma > 1)$ and resample from scratch $(i = 1, \dots M^{(k)})$. This gives a more robust algorithm with only little extra cost:

$$\text{total sample size} \quad \mathcal{M}^{(k)} = \sum_{i=0}^{k} M^{(k)} = \sum_{i=0}^{k} M^{(0)}\gamma^k \leq \frac{\gamma}{\gamma-1} M^{(k)}.$$

Complexity of Monte Carlo algorithm

Assumptions: for a d-dimensional problem

- $|\mathbb{E}[Q - Q_h]| \leq C_\alpha h^\alpha$, (grid convergence with rate α on the mean)
- $\mathbb{V}\mathrm{ar}[Q_h] \leq C_\beta$, ($\mathbb{V}\mathrm{ar}[Q_h] \approx \mathbb{V}\mathrm{ar}[Q] \nrightarrow 0$ as $h \to 0$)
- cost to compute each $Q_h^{(i)}$: $C_h \leq C_\gamma h^{-d\gamma}$

(typically, $\#dofs \simeq h^{-d}$ and the cost C_h depends algebraically on $\#dofs$)

Balancing error contributions to have $MSE \leq \mathrm{tol}^2$

$$B^2 \leq \frac{\mathrm{tol}^2}{2} \quad \Longrightarrow \quad h \simeq \mathrm{tol}^{\frac{1}{\alpha}}$$

$$V_{MC} \leq \frac{\mathrm{tol}^2}{2} \quad \Longrightarrow \quad M \simeq \mathrm{tol}^{-2}$$

Complexity analysis (error versus cost)

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) = \mathrm{C_h M} \simeq \mathrm{tol}^{-2-\frac{d\gamma}{\alpha}}$$

Complexity of Monte Carlo algorithm

Assumptions: for a d-dimensional problem

- $|\mathbb{E}[Q - Q_h]| \leq C_\alpha h^\alpha$, (grid convergence with rate α on the mean)
- $\mathbb{V}\mathrm{ar}[Q_h] \leq C_\beta$, ($\mathbb{V}\mathrm{ar}[Q_h] \approx \mathbb{V}\mathrm{ar}[Q] \nrightarrow 0$ as $h \to 0$)
- cost to compute each $Q_h^{(i)}$: $C_h \leq C_\gamma h^{-d\gamma}$

(typically, $\#dofs \simeq h^{-d}$ and the cost C_h depends algebraically on $\#dofs$)

Balancing error contributions to have $MSE \leq \mathrm{tol}^2$

$$B^2 \leq \frac{\mathrm{tol}^2}{2} \quad \Longrightarrow \quad h \simeq \mathrm{tol}^{\frac{1}{\alpha}}$$

$$V_{MC} \leq \frac{\mathrm{tol}^2}{2} \quad \Longrightarrow \quad M \simeq \mathrm{tol}^{-2}$$

Complexity analysis (error versus cost)

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) = \mathrm{C_h M} \simeq \mathrm{tol}^{-2-\frac{d\gamma}{\alpha}}$$

Complexity of Monte Carlo algorithm

Assumptions: for a *d*-dimensional problem

- $|\mathbb{E}[Q - Q_h]| \leq C_\alpha h^\alpha$, (grid convergence with rate α on the mean)
- $\mathbb{V}\mathrm{ar}[Q_h] \leq C_\beta$, ($\mathbb{V}\mathrm{ar}[Q_h] \approx \mathbb{V}\mathrm{ar}[Q] \nrightarrow 0$ as $h \to 0$)
- cost to compute each $Q_h^{(i)}$: $C_h \leq C_\gamma h^{-d\gamma}$

(typically, $\#dofs \simeq h^{-d}$ and the cost C_h depends algebraically on $\#dofs$)

Balancing error contributions to have $MSE \leq \mathrm{tol}^2$

$$B^2 \leq \frac{\mathrm{tol}^2}{2} \quad \Longrightarrow \quad h \simeq \mathrm{tol}^{\frac{1}{\alpha}}$$

$$V_{MC} \leq \frac{\mathrm{tol}^2}{2} \quad \Longrightarrow \quad M \simeq \mathrm{tol}^{-2}$$

Complexity analysis (error versus cost)

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) = \mathrm{C_h M} \simeq \mathrm{tol}^{-2-\frac{d\gamma}{\alpha}}$$

Example 1 – stochastic differential equation

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t(y), \qquad t \in (0, T], \qquad X_0 = x_0$$

Quantity of interest: $Q = \tilde{Q}(X_T)$

here $W_t(y)$ is a standard Wiener process (y denotes a random elementary event)

Discretization by Euler-Maruyama with step size $h = T/N$ and $t_n = nh$

$$X^{n+1} = X^n + a(t_n, X^n)h + b(t_n, X^n)\Delta W_n, \quad n = 0, \dots, N-1, \quad \Delta W_n \overset{iid}{\sim} N(0, h)$$

approximate quantity of interest: $Q_h = \tilde{Q}(X^N)$

For smooth $a(\cdot)$ and $b(\cdot)$ one has

- $|\mathbb{E}[Q - Q_h]| \lesssim h$ (order 1 convergence for the mean – weak rate)
- $\mathbb{E}[(Q - Q_h)^2]^{\frac{1}{2}} \lesssim h^{\frac{1}{2}}$ (order 1/2 in mean square sense – strong rate)
- $C_h = N \lesssim h^{-1}$ (cost proportional to the numebr of iterations)

Hence: $\alpha = 1$, $d = 1$, $\gamma = 1$ $\implies$ $\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3}$

To reduce the error by a factor 10, the cost increses by a factor 10^3 !

Example 1 – stochastic differential equation

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t(y), \qquad t \in (0, T], \qquad X_0 = x_0$$

Quantity of interest: $Q = \tilde{Q}(X_T)$

here $W_t(y)$ is a standard Wiener process (y denotes a random elementary event)

Discretization by Euler-Maruyama with step size $h = T/N$ and $t_n = nh$

$$X^{n+1} = X^n + a(t_n, X^n)h + b(t_n, X^n)\Delta W_n, \quad n = 0, \dots, N-1, \quad \Delta W_n \overset{iid}{\sim} N(0, h)$$

approximate quantity of interest: $Q_h = \tilde{Q}(X^N)$

For smooth $a(\cdot)$ and $b(\cdot)$ one has

- $|\mathbb{E}[Q - Q_h]| \lesssim h$ (order 1 convergence for the mean – weak rate)
- $\mathbb{E}[(Q - Q_h)^2]^{\frac{1}{2}} \lesssim h^{\frac{1}{2}}$ (order 1/2 in mean square sense – strong rate)
- $C_h = N \lesssim h^{-1}$ (cost proportional to the numebr of iterations)

Hence: $\alpha = 1$, $d = 1$, $\gamma = 1$ $\implies$ $\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3}$

To reduce the error by a factor 10, the cost increses by a factor 10^3 !

Example 1 – stochastic differential equation

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t(y), \qquad t \in (0, T], \qquad X_0 = x_0$$

Quantity of interest: $Q = \tilde{Q}(X_T)$

here $W_t(y)$ is a standard Wiener process (y denotes a random elementary event)

Discretization by Euler-Maruyama with step size $h = T/N$ and $t_n = nh$

$$X^{n+1} = X^n + a(t_n, X^n)h + b(t_n, X^n)\Delta W_n, \quad n = 0, \ldots, N-1, \quad \Delta W_n \overset{iid}{\sim} N(0, h)$$

approximate quantity of interest: $Q_h = \tilde{Q}(X^N)$

For smooth $a(\cdot)$ and $b(\cdot)$ one has

- $|\mathbb{E}[Q - Q_h]| \lesssim h$ (order 1 convergence for the mean – weak rate)
- $\mathbb{E}[(Q - Q_h)^2]^{\frac{1}{2}} \lesssim h^{\frac{1}{2}}$ (order 1/2 in mean square sense – strong rate)
- $C_h = N \lesssim h^{-1}$ (cost proportional to the numebr of iterations)

Hence: $\alpha = 1$, $d = 1$, $\gamma = 1$ $\implies$ $\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3}$

To reduce the error by a factor 10, the cost increses by a factor 10^3 !

Example 1 – stochastic differential equation

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t(y), \qquad t \in (0, T], \qquad X_0 = x_0$$

Quantity of interest: $Q = \tilde{Q}(X_T)$

here $W_t(y)$ is a standard Wiener process (y denotes a random elementary event)

Discretization by Euler-Maruyama with step size $h = T/N$ and $t_n = nh$

$$X^{n+1} = X^n + a(t_n, X^n)h + b(t_n, X^n)\Delta W_n, \quad n = 0, \dots, N-1, \quad \Delta W_n \overset{iid}{\sim} N(0, h)$$

approximate quantity of interest: $Q_h = \tilde{Q}(X^N)$

For smooth $a(\cdot)$ and $b(\cdot)$ one has

- $|\mathbb{E}[Q - Q_h]| \lesssim h$ (order 1 convergence for the mean – weak rate)
- $\mathbb{E}[(Q - Q_h)^2]^{\frac{1}{2}} \lesssim h^{\frac{1}{2}}$ (order 1/2 in mean square sense – strong rate)
- $C_h = N \lesssim h^{-1}$ (cost proportional to the numebr of iterations)

Hence: $\alpha = 1$, $d = 1$, $\gamma = 1 \implies \mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3}$

To reduce the error by a factor 10, the cost increses by a factor 10^3 !

Example 2 – PDE with random parameters

$$-\operatorname{div}(a(y)\nabla u) = f, \quad \text{in } D \subset \mathbb{R}^d, \qquad u = 0, \text{ on } \partial D$$

with $a(y)$ uniformly bounded, positive and Lipschitz continuous random field.

Quantity of interest: Lipschitz functional $Q = \tilde{Q}(u)$ (e.g. $Q = \frac{1}{|\Sigma|}\int_\Sigma \|\nabla u\|$, $\Sigma \subset D$)

Discretization by $\mathbb{P}^1$ finite elements on a regular triangulation with mesh size h.

Assumptions

- $0 < a_{min} \leq a(x, y) \leq a_{max}, \quad \forall x \in D, \ y \in \Gamma$
- $\|\nabla a(\cdot, y)\|_{L^\infty(D)} \leq K, \quad \forall y \in \Gamma$
- D Lipschitz convex, f smooth

Example 2 – PDE with random parameters

$$-\operatorname{div}(a(y)\nabla u) = f, \quad \text{in } D \subset \mathbb{R}^d, \qquad u = 0, \ \text{on } \partial D$$

with $a(y)$ uniformly bounded, positive and Lipschitz continuous random field.

Quantity of interest: Lipschitz functional $Q = \tilde{Q}(u)$ (e.g. $Q = \frac{1}{|\Sigma|}\int_\Sigma \|\nabla u\|$, $\Sigma \subset D$)

Discretization by $\mathbb{P}^1$ finite elements on a regular triangulation with mesh size h.

Assumptions

- $0 < a_{min} \leq a(x,y) \leq a_{max}, \ \ \forall x \in D, \ y \in \Gamma$
- $\|\nabla a(\cdot, y)\|_{L^\infty(D)} \leq K, \ \ \forall y \in \Gamma$
- D Lipschitz convex, f smooth

Example 2 – PDE with random parameters

$$-\operatorname{div}(a(y)\nabla u) = f, \quad \text{in } D \subset \mathbb{R}^d, \qquad u = 0, \ \text{on } \partial D$$

with $a(y)$ uniformly bounded, positive and Lipschitz continuous random field.

Quantity of interest: Lipschitz functional $Q = \tilde{Q}(u)$ (e.g. $Q = \frac{1}{|\Sigma|}\int_\Sigma \|\nabla u\|$, $\Sigma \subset D$)

Discretization by $\mathbb{P}^1$ finite elements on a regular triangulation with mesh size h.

Assumptions

- $0 < a_{min} \leq a(x,y) \leq a_{max}, \ \ \forall x \in D, \ y \in \Gamma$
- $\|\nabla a(\cdot, y)\|_{L^\infty(D)} \leq K, \ \ \forall y \in \Gamma$
- D Lipschitz convex, f smooth

Example 2 – PDE with random parameters

Then

1. $\|u(y) - u_h(y)\|_{H^1} \leq Ch$, $\forall y \in \Gamma$ (order 1 "pathwise" convergence rate)
2. $|Q(y) - Q_h(y)| \lesssim \|u(y) - u_h(y)\|_{H^1} \lesssim h$, $\forall y \in \Gamma$ (order 1 "pathwise" convergence on Lipschitz functionals; for smoother functionals the rate could be up to 2)
3. $C_h \lesssim (\#dofs)^\gamma \lesssim h^{-d\gamma}$
   Here γ denotes the complexity to solve the linear system ($\gamma = 3$ for a direct (full) solver; $\gamma \approx 1$ for an iterative method with optimal preconditioner)

From 2. we deduce $|\mathbb{E}[Q - Q_h]| \leq \mathbb{E}[|Q - Q_h|] \lesssim h$.

Hence $\alpha = 1$ and $\gamma = 1$ (optimal solver). For a 3D problem $d = 3$

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-5}$$

To reduce the error by a factor 10, the cost increses by a factor 10^5 !

Can we do better than that ?

Yes. Multilevel Monte Carlo can bring this cost down to tol^{-2} in favorable cases !

Example 2 – PDE with random parameters

Then

1. $\|u(y) - u_h(y)\|_{H^1} \leq Ch, \ \forall y \in \Gamma$ (order 1 "pathwise" convergence rate)
2. $|Q(y) - Q_h(y)| \lesssim \|u(y) - u_h(y)\|_{H^1} \lesssim h, \ \forall y \in \Gamma$ (order 1 "pathwise" convergence on Lipschitz functionals; for smoother functionals the rate could be up to 2)
3. $C_h \lesssim (\#dofs)^\gamma \lesssim h^{-d\gamma}$
   Here γ denotes the complexity to solve the linear system ($\gamma = 3$ for a direct (full) solver; $\gamma \approx 1$ for an iterative method with optimal preconditioner)

From 2. we deduce $|\mathbb{E}[Q - Q_h]| \leq \mathbb{E}[|Q - Q_h|] \lesssim h$.

Hence $\alpha = 1$ and $\gamma = 1$ (optimal solver). For a 3D problem $d = 3$

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-5}$$

To reduce the error by a factor 10, the cost increses by a factor 10^5 !

Can we do better than that ?

Yes. Multilevel Monte Carlo can bring this cost down to tol^{-2} in favorable cases !

Example 2 – PDE with random parameters

Then

1. $\|u(y) - u_h(y)\|_{H^1} \leq Ch, \ \forall y \in \Gamma$ (order 1 "pathwise" convergence rate)
2. $|Q(y) - Q_h(y)| \lesssim \|u(y) - u_h(y)\|_{H^1} \lesssim h, \ \forall y \in \Gamma$ (order 1 "pathwise" convergence on Lipschitz functionals; for smoother functionals the rate could be up to 2)
3. $C_h \lesssim (\#dofs)^{\gamma} \lesssim h^{-d\gamma}$
   Here γ denotes the complexity to solve the linear system ($\gamma = 3$ for a direct (full) solver; $\gamma \approx 1$ for an iterative method with optimal preconditioner)

From 2. we deduce $|\mathbb{E}[Q - Q_h]| \leq \mathbb{E}[|Q - Q_h|] \lesssim h$.

Hence $\alpha = 1$ and $\gamma = 1$ (optimal solver). For a 3D problem $d = 3$

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-5}$$

To reduce the error by a factor 10, the cost increses by a factor 10^5 !

Can we do better than that ?

Yes. Multilevel Monte Carlo can bring this cost down to tol^{-2} in favorable cases !

Example 2 – PDE with random parameters

Then

1. $\|u(y) - u_h(y)\|_{H^1} \leq Ch$, $\forall y \in \Gamma$ (order 1 "pathwise" convergence rate)
2. $|Q(y) - Q_h(y)| \lesssim \|u(y) - u_h(y)\|_{H^1} \lesssim h$, $\forall y \in \Gamma$ (order 1 "pathwise" convergence on Lipschitz functionals; for smoother functionals the rate could be up to 2)
3. $C_h \lesssim (\#dofs)^\gamma \lesssim h^{-d\gamma}$
   Here γ denotes the complexity to solve the linear system ($\gamma = 3$ for a direct (full) solver; $\gamma \approx 1$ for an iterative method with optimal preconditioner)

From 2. we deduce $|\mathbb{E}[Q - Q_h]| \leq \mathbb{E}[|Q - Q_h|] \lesssim h$.

Hence $\alpha = 1$ and $\gamma = 1$ (optimal solver). For a 3D problem $d = 3$

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-5}$$

To reduce the error by a factor 10, the cost increses by a factor 10^5 !

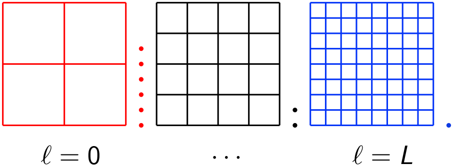
Can we do better than that ?

Yes. Multilevel Monte Carlo can bring this cost down to tol^{-2} in favorable cases !

Outline

Multilevel Monte Carlo (MLMC) method

Iterated control variate idea [Heinrich 1998], [Giles 2008]



- Sequence of refined discretizations (not necessarily nested nor structured)

$$h_0 > h_1 > \ldots > h_L$$

- Sequence of sample sizes

$$M_0 > M_1 > \cdots > M_L$$

We assume that the mesh size h_L achieves the desired accuracy and aim at computing $\mathbb{E}[Q_{h_L}]$.

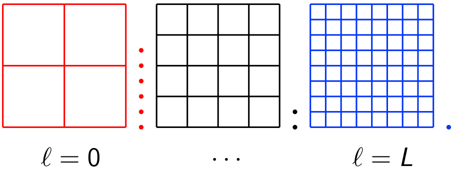
Simple idea: write a **telescopic sum** (denoting $Q_\ell = Q_{h_\ell}$)

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \mathbb{E}[Q_1 - Q_0] + \ldots + \mathbb{E}[Q_L - Q_{L-1}]$$

and estimate each term independently with different sample sizes

Multilevel Monte Carlo (MLMC) method

Iterated control variate idea [Heinrich 1998], [Giles 2008]



- Sequence of refined discretizations (not necessarily nested nor structured)

$$h_0 > h_1 > \ldots > h_L$$

- Sequence of sample sizes

$$M_0 > M_1 > \cdots > M_L$$

We assume that the mesh size h_L achieves the desired accuracy and aim at computing $\mathbb{E}[Q_{h_L}]$.

Simple idea: write a **telescopic sum** (denoting $Q_\ell = Q_{h_\ell}$)

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \mathbb{E}[Q_1 - Q_0] + \ldots + \mathbb{E}[Q_L - Q_{L-1}]$$

and estimate each term independently with different sample sizes

MLMC estimator

$$\hat{\mu}_{MLMC} = \hat{\mathbb{E}}_{M_0}[Q_0] + \hat{\mathbb{E}}_{M_1}[Q_1 - Q_0] + \ldots + \hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]$$

$$= \sum_{\ell=0}^{L} \frac{1}{M_\ell} \sum_{i=1}^{M_\ell} (Q_\ell(y^{(i,\ell)}) - Q_{\ell-1}(y^{(i,\ell)})), \qquad Q_{-1} = 0$$

Notice that $Q_\ell(y^{(i,\ell)})$ and $Q_{\ell-1}(y^{(i,\ell)})$ are evaluated for the same realization of the random varaibles $y^{(i,\ell)}$. Hence, the difference is hopefully small for large ℓ as $Q_\ell(y) \xrightarrow{h\to 0} Q(y)$.

Bias (discretization error):

$$\mathbb{E}[\hat{\mu}_{MLMC}] - \mu = \sum_{\ell=0}^{L} \mathbb{E}\left[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\right] - \mu = \sum_{\ell=0}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] - \mu$$

$$= \mathbb{E}[Q_L] - \mu$$

Notice that the bias depens only on the finest discretization level – controlled by the choice of h_L.

MLMC estimator

$$\hat{\mu}_{MLMC} = \hat{\mathbb{E}}_{M_0}[Q_0] + \hat{\mathbb{E}}_{M_1}[Q_1 - Q_0] + \ldots + \hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]$$
$$= \sum_{\ell=0}^{L} \frac{1}{M_\ell} \sum_{i=1}^{M_\ell} (Q_\ell(y^{(i,\ell)}) - Q_{\ell-1}(y^{(i,\ell)})), \qquad Q_{-1} = 0$$

Notice that $Q_\ell(y^{(i,\ell)})$ and $Q_{\ell-1}(y^{(i,\ell)})$ are evaluated for the same realization of the random varaibles $y^{(i,\ell)}$. Hence, the difference is hopefully small for large ℓ as $Q_\ell(y) \xrightarrow{h \to 0} Q(y)$.

Bias (discretization error):

$$\mathbb{E}[\hat{\mu}_{MLMC}] - \mu = \sum_{\ell=0}^{L} \mathbb{E}\left[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\right] - \mu = \sum_{\ell=0}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] - \mu$$
$$= \mathbb{E}[Q_L] - \mu$$

Notice that the bias depens only on the finest discretization level – controlled by the choice of h_L.

MLMC estimator

$$\hat{\mu}_{MLMC} = \hat{\mathbb{E}}_{M_0}[Q_0] + \hat{\mathbb{E}}_{M_1}[Q_1 - Q_0] + \ldots + \hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]$$
$$= \sum_{\ell=0}^{L} \frac{1}{M_\ell} \sum_{i=1}^{M_\ell} (Q_\ell(y^{(i,\ell)}) - Q_{\ell-1}(y^{(i,\ell)})), \qquad Q_{-1} = 0$$

Notice that $Q_\ell(y^{(i,\ell)})$ and $Q_{\ell-1}(y^{(i,\ell)})$ are evaluated for the same realization of the random varaibles $y^{(i,\ell)}$. Hence, the difference is hopefully small for large ℓ as $Q_\ell(y) \xrightarrow{h \to 0} Q(y)$.

Bias (discretization error):

$$\mathbb{E}[\hat{\mu}_{MLMC}] - \mu = \sum_{\ell=0}^{L} \mathbb{E}\left[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\right] - \mu = \sum_{\ell=0}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] - \mu$$
$$= \mathbb{E}[Q_L] - \mu$$

Notice that the bias depens only on the finest discretization level – controlled by the choice of h_L.

MLMC estimator

$$\hat{\mu}_{MLMC} = \hat{\mathbb{E}}_{M_0}[Q_0] + \hat{\mathbb{E}}_{M_1}[Q_1 - Q_0] + \ldots + \hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]$$

$$= \sum_{\ell=0}^{L} \frac{1}{M_\ell} \sum_{i=1}^{M_\ell} (Q_\ell(y^{(i,\ell)}) - Q_{\ell-1}(y^{(i,\ell)})), \qquad Q_{-1} = 0$$

Notice that $Q_\ell(y^{(i,\ell)})$ and $Q_{\ell-1}(y^{(i,\ell)})$ are evaluated for the same realization of the random varaibles $y^{(i,\ell)}$. Hence, the difference is hopefully small for large ℓ as $Q_\ell(y) \xrightarrow{h \to 0} Q(y)$.

Bias (discretization error):

$$\mathbb{E}[\hat{\mu}_{MLMC}] - \mu = \sum_{\ell=0}^{L} \mathbb{E}\left[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\right] - \mu = \sum_{\ell=0}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] - \mu$$

$$= \mathbb{E}[Q_L] - \mu$$

Notice that the bias depens only on the finest discretization level – controlled by the choice of h_L.

MLMC estimator

$$\hat{\mu}_{MLMC} = \hat{\mathbb{E}}_{M_0}[Q_0] + \hat{\mathbb{E}}_{M_1}[Q_1 - Q_0] + \ldots + \hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]$$
$$= \sum_{\ell=0}^{L} \frac{1}{M_\ell} \sum_{i=1}^{M_\ell} (Q_\ell(y^{(i,\ell)}) - Q_{\ell-1}(y^{(i,\ell)})), \qquad Q_{-1} = 0$$

Notice that $Q_\ell(y^{(i,\ell)})$ and $Q_{\ell-1}(y^{(i,\ell)})$ are evaluated for the same realization of the random varaibles $y^{(i,\ell)}$. Hence, the difference is hopefully small for large ℓ as $Q_\ell(y) \xrightarrow{h\to 0} Q(y)$.

Bias (discretization error):

$$\mathbb{E}[\hat{\mu}_{MLMC}] - \mu = \sum_{\ell=0}^{L} \mathbb{E}\left[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\right] - \mu = \sum_{\ell=0}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] - \mu$$
$$= \mathbb{E}[Q_L] - \mu$$

Notice that the bias depens only on the finest discretization level – controlled by the choice of h_L.

MLMC estimator

$$\hat{\mu}_{MLMC} = \hat{\mathbb{E}}_{M_0}[Q_0] + \hat{\mathbb{E}}_{M_1}[Q_1 - Q_0] + \ldots + \hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]$$
$$= \sum_{\ell=0}^{L} \frac{1}{M_\ell} \sum_{i=1}^{M_\ell} (Q_\ell(y^{(i,\ell)}) - Q_{\ell-1}(y^{(i,\ell)})), \qquad Q_{-1} = 0$$

Notice that $Q_\ell(y^{(i,\ell)})$ and $Q_{\ell-1}(y^{(i,\ell)})$ are evaluated for the same realization of the random varaibles $y^{(i,\ell)}$. Hence, the difference is hopefully small for large ℓ as $Q_\ell(y) \xrightarrow{h\to 0} Q(y)$.

Bias (discretization error):

$$\mathbb{E}[\hat{\mu}_{MLMC}] - \mu = \sum_{\ell=0}^{L} \mathbb{E}\left[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\right] - \mu = \sum_{\ell=0}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] - \mu$$
$$= \mathbb{E}[Q_L] - \mu$$

Notice that the bias depens only on the finest discretization level – controlled by the choice of h_L.

MLMC estimator

Variance (statistical error):

$$V_{MLMC} = \mathbb{V}\mathrm{ar}\Big[\sum_{\ell=0}^{L} \hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\Big] = \sum_{\ell=0}^{L} \mathbb{V}\mathrm{ar}\big[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\big]$$

$$= \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

Controlled by the choice of sample sizes $\{M_\ell\}_{\ell=0}^{L}$.

Mean squared error

$$\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

- **Key point**: Since $\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$ gets smaller and smaller for large ℓ, one can take M_ℓ smaller and smaller. Only few samples on the fine grid h_L.
- The level 0 is usually determined by stability and accuracy requirements. In particular, one needs $\mathbb{V}\mathrm{ar}[Q_1 - Q_0] \ll \mathbb{V}\mathrm{ar}[Q_0] \approx \mathbb{V}\mathrm{ar}[Q]$.

MLMC estimator

Variance (statistical error):

$$V_{MLMC} = \mathbb{V}\mathrm{ar}\Big[\sum_{\ell=0}^{L} \hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\Big] = \sum_{\ell=0}^{L} \mathbb{V}\mathrm{ar}\big[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\big]$$

$$= \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

Controlled by the choice of sample sizes $\{M_\ell\}_{\ell=0}^{L}$.

Mean squared error

$$\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

- **Key point**: Since $\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$ gets smaller and smaller for large ℓ, one can take M_ℓ smaller and smaller. Only few samples on the fine grid h_L.
- The level 0 is usually determined by stability and accuracy requirements. In particular, one needs $\mathbb{V}\mathrm{ar}[Q_1 - Q_0] \ll \mathbb{V}\mathrm{ar}[Q_0] \approx \mathbb{V}\mathrm{ar}[Q]$.

MLMC estimator

Variance (statistical error):

$$V_{MLMC} = \mathbb{V}\mathrm{ar}\Big[\sum_{\ell=0}^{L} \hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\Big] = \sum_{\ell=0}^{L} \mathbb{V}\mathrm{ar}\big[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\big]$$

$$= \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

Controlled by the choice of sample sizes $\{M_\ell\}_{\ell=0}^{L}$.

Mean squared error

$$\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

- **Key point**: Since $\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$ gets smaller and smaller for large ℓ, one can take M_ℓ smaller and smaller. Only few samples on the fine grid h_L.
- The level 0 is usually determined by stability and accuracy requirements. In particular, one needs $\mathbb{V}\mathrm{ar}[Q_1 - Q_0] \ll \mathbb{V}\mathrm{ar}[Q_0] \approx \mathbb{V}\mathrm{ar}[Q]$.

MLMC estimator

Variance (statistical error):

$$V_{MLMC} = \mathbb{V}\mathrm{ar}\Big[\sum_{\ell=0}^{L} \hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\Big] = \sum_{\ell=0}^{L} \mathbb{V}\mathrm{ar}\big[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\big]$$

$$= \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

Controlled by the choice of sample sizes $\{M_\ell\}_{\ell=0}^{L}$.

Mean squared error

$$\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

- **Key point**: Since $\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$ gets smaller and smaller for large ℓ, one can take M_ℓ smaller and smaller. Only few samples on the fine grid h_L.
- The level 0 is usually determined by stability and accuracy requirements. In particular, one needs $\mathbb{V}\mathrm{ar}[Q_1 - Q_0] \ll \mathbb{V}\mathrm{ar}[Q_0] \approx \mathbb{V}\mathrm{ar}[Q]$.

MLMC estimator

Variance (statistical error):

$$V_{MLMC} = \mathbb{V}\mathrm{ar}\Big[\sum_{\ell=0}^{L} \hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\Big] = \sum_{\ell=0}^{L} \mathbb{V}\mathrm{ar}\big[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\big]$$

$$= \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

Controlled by the choice of sample sizes $\{M_\ell\}_{\ell=0}^{L}$.

Mean squared error

$$\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

- **Key point**: Since $\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$ gets smaller and smaller for large ℓ, one can take M_ℓ smaller and smaller. Only few samples on the fine grid h_L.
- The level 0 is usually determined by stability and accuracy requirements. In particular, one needs $\mathbb{V}\mathrm{ar}[Q_1 - Q_0] \ll \mathbb{V}\mathrm{ar}[Q_0] \approx \mathbb{V}\mathrm{ar}[Q]$.

MLMC estimator

Variance (statistical error):

$$V_{MLMC} = \mathbb{V}\mathrm{ar}\Big[\sum_{\ell=0}^{L} \hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\Big] = \sum_{\ell=0}^{L} \mathbb{V}\mathrm{ar}\big[\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]\big]$$

$$= \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

Controlled by the choice of sample sizes $\{M_\ell\}_{\ell=0}^{L}$.

Mean squared error

$$\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \frac{\mathbb{V}\mathrm{ar}[Q_0]}{M_0} + \sum_{\ell=1}^{L} \frac{\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]}{M_\ell}$$

- **Key point**: Since $\mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$ gets smaller and smaller for large ℓ, one can take M_ℓ smaller and smaller. Only few samples on the fine grid h_L.
- The level 0 is usually determined by stability and accuracy requirements. In particular, one needs $\mathbb{V}\mathrm{ar}[Q_1 - Q_0] \ll \mathbb{V}\mathrm{ar}[Q_0] \approx \mathbb{V}\mathrm{ar}[Q]$.

Optimal choice of M_ℓ (optimal allocation)

- C_0: cost of generating one realization of Q_0
- C_ℓ: cost of generating one realization of $Q_\ell - Q_{\ell-1}$, $\ell > 0$
- $V_0 = \mathbb{V}\mathrm{ar}[Q_0]$
- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$, $\ell > 0$

Then

Total cost: $\mathrm{Cost}(\hat{\mu}_{MLMC}) = \sum_{\ell=0}^{L} M_\ell C_\ell$, Total variance: $\mathbb{V}\mathrm{ar}[\hat{\mu}_{MLMC}] = \sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}$.

Problem: Find optimal $\{M_\ell\}_{\ell=0}^{L}$ to minimize the cost at a fixed variance level

$$\min_{\{M_\ell\}} \sum_{\ell=0}^{L} M_\ell C_\ell \qquad \text{subject to} \quad \sum_{\ell=0}^{L} M_\ell^{-1} V_\ell \leq \mathrm{tol}^2$$

Solution: if we replace M_ℓ by continuous variables (relaxation), the optimal solution is

$$M_\ell = \mathrm{tol}^{-2} \sqrt{\frac{V_\ell}{C_\ell}} \sum_{j=0}^{L} \sqrt{V_j C_j}$$

Optimal choice of M_ℓ (optimal allocation)

- C_0: cost of generating one realization of Q_0
- C_ℓ: cost of generating one realization of $Q_\ell - Q_{\ell-1}$, $\ell > 0$
- $V_0 = \mathbb{V}\mathrm{ar}[Q_0]$
- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$, $\ell > 0$

Then

Total cost: $\mathrm{Cost}(\hat{\mu}_{MLMC}) = \sum_{\ell=0}^{L} M_\ell C_\ell,$ Total variance: $\mathbb{V}\mathrm{ar}[\hat{\mu}_{MLMC}] = \sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}.$

Problem: Find optimal $\{M_\ell\}_{\ell=0}^{L}$ to minimize the cost at a fixed variance level

$$\min_{\{M_\ell\}} \sum_{\ell=0}^{L} M_\ell C_\ell \qquad \text{subject to} \quad \sum_{\ell=0}^{L} M_\ell^{-1} V_\ell \leq \mathrm{tol}^2$$

Solution: if we replace M_ℓ by continuous variables (relaxation), the optimal solution is

$$M_\ell = \mathrm{tol}^{-2} \sqrt{\frac{V_\ell}{C_\ell}} \sum_{j=0}^{L} \sqrt{V_j C_j}$$

Optimal choice of M_ℓ (optimal allocation)

- C_0: cost of generating one realization of Q_0
- C_ℓ: cost of generating one realization of $Q_\ell - Q_{\ell-1}$, $\ell > 0$
- $V_0 = \mathbb{V}\mathrm{ar}[Q_0]$
- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$, $\ell > 0$

Then

Total cost: $\mathrm{Cost}(\hat{\mu}_{MLMC}) = \sum_{\ell=0}^{L} M_\ell C_\ell,$ Total variance: $\mathbb{V}\mathrm{ar}[\hat{\mu}_{MLMC}] = \sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}.$

Problem: Find optimal $\{M_\ell\}_{\ell=0}^{L}$ to minimize the cost at a fixed variance level

$$\min_{\{M_\ell\}} \sum_{\ell=0}^{L} M_\ell C_\ell \qquad \text{subject to} \quad \sum_{\ell=0}^{L} M_\ell^{-1} V_\ell \le \mathrm{tol}^2$$

Solution: if we replace M_ℓ by continuous variables (relaxation), the optimal solution is

$$M_\ell = \mathrm{tol}^{-2} \sqrt{\frac{V_\ell}{C_\ell}} \sum_{j=0}^{L} \sqrt{V_j C_j}$$

Proof.

Define the Lagrangian function

$$\mathcal{L}(M_0, \ldots, M_L, \lambda) = \sum_{\ell=0}^{L} M_\ell C_\ell - \lambda \left(\mathrm{tol}^2 - \sum_{j=0}^{L} \frac{V_j}{M_j} \right)$$

Then

$$\frac{\partial \mathcal{L}}{\partial M_\ell} = C_\ell - \lambda \frac{V_\ell}{M_\ell^2} = 0, \qquad \Longrightarrow \qquad M_\ell = \sqrt{\lambda \frac{V_\ell}{C_\ell}}$$

Substituting into the constraint gives

$$\sum_{j=0}^{L} \sqrt{\frac{V_j C_j}{\lambda}} = \mathrm{tol}^2, \quad \Longrightarrow \quad \sqrt{\lambda} = \mathrm{tol}^{-2} \sum_{j=0}^{L} \sqrt{V_j C_j}$$

Proof.

Define the Lagrangian function

$$\mathcal{L}(M_0,\dots,M_L,\lambda)=\sum_{\ell=0}^{L} M_\ell C_\ell-\lambda\left(\mathrm{tol}^2-\sum_{\mathrm{j}=0}^{\mathrm{L}}\frac{\mathrm{V_j}}{\mathrm{M_j}}\right)$$

Then

$$\frac{\partial\mathcal{L}}{\partial M_\ell}=C_\ell-\lambda\frac{V_\ell}{M_\ell^2}=0, \qquad\Longrightarrow\qquad M_\ell=\sqrt{\lambda\frac{V_\ell}{C_\ell}}$$

Substituting into the constraint gives

$$\sum_{j=0}^{L}\sqrt{\frac{V_jC_j}{\lambda}}=\mathrm{tol}^2, \qquad\Longrightarrow\qquad \sqrt{\lambda}=\mathrm{tol}^{-2}\sum_{\mathrm{j}=0}^{\mathrm{L}}\sqrt{\mathrm{V_jC_j}}$$

□

Proof.

Define the Lagrangian function

$$\mathcal{L}(M_0,\dots,M_L,\lambda) = \sum_{\ell=0}^{L} M_\ell C_\ell - \lambda \left(\mathrm{tol}^2 - \sum_{j=0}^{L} \frac{V_j}{M_j} \right)$$

Then

$$\frac{\partial \mathcal{L}}{\partial M_\ell} = C_\ell - \lambda \frac{V_\ell}{M_\ell^2} = 0, \qquad \Longrightarrow \qquad M_\ell = \sqrt{\lambda \frac{V_\ell}{C_\ell}}$$

Substituting into the constraint gives

$$\sum_{j=0}^{L} \sqrt{\frac{V_j C_j}{\lambda}} = \mathrm{tol}^2, \quad \Longrightarrow \quad \sqrt{\lambda} = \mathrm{tol}^{-2} \sum_{j=0}^{L} \sqrt{V_j C_j}$$

□

In practice, one should take the ceiling of the real value M_ℓ (important if $M_\ell < 1$). That is, we have for the MLMC estimator

- Optimal sample sizes: $M_\ell = \left\lceil \mathrm{tol}^{-2} \sqrt{\frac{\mathrm{V}_\ell}{\mathrm{C}_\ell}} \sum_{\mathrm{j}=0}^{\mathrm{L}} \sqrt{\mathrm{V_j C_j}} \right\rceil$

Replacing in the cost extression $\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) = \sum_{\ell=0}^{\mathrm{L}} \mathrm{C}_\ell \mathrm{M}_\ell$ and using that $\lceil x \rceil \leq x + 1$, $\forall x \in \mathbb{R}$

- Optimal work: $\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{tol}^{-2} \left(\sum_{\mathrm{j}=0}^{\mathrm{L}} \sqrt{\mathrm{V_j C_j}} \right)^2 + \sum_{\ell=0}^{\mathrm{L}} \mathrm{C}_\ell$

In practice, one should take the ceiling of the real value M_ℓ (important if $M_\ell < 1$). That is, we have for the MLMC estimator

- Optimal sample sizes: $M_\ell = \left\lceil \mathrm{tol}^{-2}\sqrt{\frac{\mathrm{V}_\ell}{\mathrm{C}_\ell}}\sum_{\mathrm{j}=0}^{\mathrm{L}}\sqrt{\mathrm{V}_\mathrm{j}\mathrm{C}_\mathrm{j}}\right\rceil$

Replacing in the cost extression $\mathrm{Cost}(\hat{\mu}_{MLMC},\mathrm{tol}) = \sum_{\ell=0}^{\mathrm{L}}\mathrm{C}_\ell\mathrm{M}_\ell$ and using that $\lceil x\rceil \leq x+1$, $\forall x \in \mathbb{R}$

- Optimal work: $\mathrm{Cost}(\hat{\mu}_{MLMC},\mathrm{tol}) \leq \mathrm{tol}^{-2}\left(\sum_{\mathrm{j}=0}^{\mathrm{L}}\sqrt{\mathrm{V}_\mathrm{j}\mathrm{C}_\mathrm{j}}\right)^2 + \sum_{\ell=0}^{\mathrm{L}}\mathrm{C}_\ell$

Complexity analysis (error vs. cost)

To analyze the complexity of the MLMC estimator, we make the following assumptions (see also [Giles 2008], [Cliffe et al. 2011])

Assumptions: for a problem in $D \subset \mathbb{R}^d$ (d-dimensional)

1. $h_\ell = h_0 \delta^\ell, \quad 0 < \delta < 1$ (sequence of geometric meshes)
2. $|\mathbb{E}[Q - Q_\ell]| \leq C_\alpha h_\ell^\alpha$ (weak rate of conv.)
3. $\mathbb{E}[(Q - Q_\ell)^2] \leq \hat{C}_\beta h_\ell^\beta$ (strong rate of conv.)
4. $C_\ell = C_\gamma h_\ell^{-d\gamma}$ ($\gamma = 3$

Notice that from 3 it follows that

6. $V_\ell \leq C_\beta h_\ell^\beta$, with $C_\beta = 2\hat{C}_\beta(1 + \delta^{-\beta})$.

Indeed:

$$V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] \leq \mathbb{E}[(Q_\ell - Q_{\ell-1})^2] \\ \leq 2\mathbb{E}[(Q - Q_\ell)^2] + 2\mathbb{E}[(Q - Q_{\ell-1})^2] \leq 2\hat{C}_\beta(1 + \delta^{-\beta}) h_\ell^\beta$$

Complexity analysis (error vs. cost)

To analyze the complexity of the MLMC estimator, we make the following assumptions (see also [Giles 2008], [Cliffe et al. 2011])

Assumptions: for a problem in $D \subset \mathbb{R}^d$ (d-dimensional)

1. $h_\ell = h_0 \delta^\ell, \quad 0 < \delta < 1$ (sequence of geometric meshes)
2. $|\mathbb{E}[Q - Q_\ell]| \leq C_\alpha h_\ell^\alpha$ (weak rate of conv.)
3. $\mathbb{E}[(Q - Q_\ell)^2] \leq \hat{C}_\beta h_\ell^\beta$ (strong rate of conv.)
4. $C_\ell = C_\gamma h_\ell^{-d\gamma}$ ($\gamma = 3$

Notice that from 3 it follows that

6. $V_\ell \leq C_\beta h_\ell^\beta$, with $C_\beta = 2\hat{C}_\beta(1 + \delta^{-\beta})$.

Indeed:

$$V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] \leq \mathbb{E}[(Q_\ell - Q_{\ell-1})^2]$$
$$\leq 2\mathbb{E}[(Q - Q_\ell)^2] + 2\mathbb{E}[(Q - Q_{\ell-1})^2] \leq 2\hat{C}_\beta(1 + \delta^{-\beta}) h_\ell^\beta$$

Complexity analysis (error vs. cost)

Moreover one always has $\beta \leq 2\alpha$ (typically $\beta = 2\alpha$ for PDEs with random coefficients). Indeed by Cauchy-Schwarz inequality

$$\mathbb{E}[Q - Q_\ell] \leq \mathbb{E}[(Q - Q_\ell)^2]^{\frac{1}{2}} \leq \sqrt{\tilde{C}_\beta} h_\ell^{\frac{\beta}{2}}, \quad \text{hence} \quad \alpha \geq \frac{\beta}{2} .$$

Theorem (MLMC Complexity, [Cliffe et al. 2011])

Under the assumptions 1-4 above, if $2\alpha \geq \min(\beta, d\gamma)$, the computational cost required to approximate $\mathbb{E}[Q]$ with MLMC with accuracy $0 < \mathrm{tol} < 1/\mathrm{e}$ in mean square sense, that is $\mathbb{E}[(\hat{\mu}_{MLMC} - \mu)^2] \leq \mathrm{tol}^2$ is bounded as follows:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{C} \begin{cases} \mathrm{tol}^{-2}, & \text{for } \beta > \mathrm{d}\gamma, \\ \mathrm{tol}^{-2} \log^2(\mathrm{tol}), & \text{for } \beta = \mathrm{d}\gamma, \\ \mathrm{tol}^{-2-(\mathrm{d}\gamma-\beta)/\alpha}, & \text{for } \beta < \mathrm{d}\gamma, \end{cases}$$

Recall: standard MC has corresponding complexity of

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \propto \mathrm{tol}^{-2-\mathrm{d}\gamma/\alpha} .$$

Complexity analysis (error vs. cost)

Moreover one always has $\beta \leq 2\alpha$ (typically $\beta = 2\alpha$ for PDEs with random coefficients). Indeed by Cauchy-Schwarz inequality

$$\mathbb{E}[Q - Q_\ell] \leq \mathbb{E}[(Q - Q_\ell)^2]^{\frac{1}{2}} \leq \sqrt{\tilde{C}_\beta} h_\ell^{\frac{\beta}{2}}, \quad \text{hence} \quad \alpha \geq \frac{\beta}{2} .$$

Theorem (MLMC Complexity, [Cliffe et al. 2011])

Under the assumptions 1-4 above, if $2\alpha \geq \min(\beta, d\gamma)$, *the computational cost required to approximate* $\mathbb{E}[Q]$ *with MLMC with accuracy* $0 < \mathrm{tol} < 1/\mathrm{e}$ *in mean square sense, that is* $\mathbb{E}[(\hat{\mu}_{MLMC} - \mu)^2] \leq \mathrm{tol}^2$ *is bounded as follows:*

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{C} \begin{cases} \mathrm{tol}^{-2}, & \text{for } \beta > \mathrm{d}\gamma, \\ \mathrm{tol}^{-2} \log^2(\mathrm{tol}), & \text{for } \beta = \mathrm{d}\gamma, \\ \mathrm{tol}^{-2-(\mathrm{d}\gamma-\beta)/\alpha}, & \text{for } \beta < \mathrm{d}\gamma, \end{cases}$$

Recall: standard MC has corresponding complexity of

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \propto \mathrm{tol}^{-2-\mathrm{d}\gamma/\alpha} .$$

Complexity analysis (error vs. cost)

Moreover one always has $\beta \leq 2\alpha$ (typically $\beta = 2\alpha$ for PDEs with random coefficients). Indeed by Cauchy-Schwarz inequality

$$\mathbb{E}[Q - Q_\ell] \leq \mathbb{E}[(Q - Q_\ell)^2]^{\frac{1}{2}} \leq \sqrt{\tilde{C}_\beta} h_\ell^{\frac{\beta}{2}}, \quad \text{hence} \quad \alpha \geq \frac{\beta}{2} \, .$$

Theorem (MLMC Complexity, [Cliffe et al. 2011])

Under the assumptions 1-4 above, if $2\alpha \geq \min(\beta, d\gamma)$, the computational cost required to approximate $\mathbb{E}[Q]$ with MLMC with accuracy $0 < \mathrm{tol} < 1/\mathrm{e}$ in mean square sense, that is $\mathbb{E}[(\hat{\mu}_{MLMC} - \mu)^2] \leq \mathrm{tol}^2$ is bounded as follows:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{C} \begin{cases} \mathrm{tol}^{-2}, & \text{for } \beta > \mathrm{d}\gamma, \\ \mathrm{tol}^{-2} \log^2(\mathrm{tol}), & \text{for } \beta = \mathrm{d}\gamma, \\ \mathrm{tol}^{-2-(\mathrm{d}\gamma - \beta)/\alpha}, & \text{for } \beta < \mathrm{d}\gamma, \end{cases}$$

Recall: standard MC has corresponding complexity of

$$\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \propto \mathrm{tol}^{-2-\mathrm{d}\gamma/\alpha} \, .$$

Proof.
We enforce the error constraint $\mathrm{MSE}(\hat{\mu}_{MLMC}) \leq \mathrm{tol}^2$ as

Bias constraint: $|\mathbb{E}[Q - Q_L]|^2 \leq \frac{1}{2}\mathrm{tol}^2$, Var. constraint: $\mathbb{V}\mathrm{ar}[\hat{\mu}_{\mathrm{MLMC}}] \leq \frac{1}{2}\mathrm{tol}^2$

From the Bias constraint we get

$$L(\mathrm{tol}) \equiv \mathrm{L} = \left\lceil \frac{\log(\sqrt{2}\mathrm{C}_\alpha \mathrm{h}_0^\alpha \mathrm{tol}^{-1})}{\alpha \log \delta^{-1}} \right\rceil \sim \log_\delta \mathrm{tol}^{\frac{1}{\alpha}} .$$

Setting $\tilde{C}_\beta = C_\beta h_0^\beta$ and $\tilde{C}_\gamma = C_\gamma h_0^{-d\gamma}$, the total cost is:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{tol}^{-2} \left(\sum_{\mathrm{j}=1}^{\mathrm{L}} \sqrt{\tilde{\mathrm{C}}_\beta \tilde{\mathrm{C}}_\gamma \delta^{\mathrm{j}\frac{\beta - \mathrm{d}\gamma}{2}}} \right)^2 + \sum_{\mathrm{j}=0}^{\mathrm{L}} \mathrm{C}_\mathrm{j}$$

Proof.

We enforce the error constraint $\mathrm{MSE}(\hat{\mu}_{MLMC}) \leq \mathrm{tol}^2$ as

Bias constraint: $|\mathbb{E}[Q - Q_L]|^2 \leq \frac{1}{2}\mathrm{tol}^2$, Var. constraint: $\mathbb{V}\mathrm{ar}[\hat{\mu}_{\mathrm{MLMC}}] \leq \frac{1}{2}\mathrm{tol}^2$

From the Bias constraint we get

$$L(\mathrm{tol}) \equiv \mathrm{L} = \left\lceil \frac{\log(\sqrt{2}\mathrm{C}_\alpha \mathrm{h}_0^\alpha \mathrm{tol}^{-1})}{\alpha \log \delta^{-1}} \right\rceil \sim \log_\delta \mathrm{tol}^{\frac{1}{\alpha}} .$$

Setting $\tilde{C}_\beta = C_\beta h_0^\beta$ and $\tilde{C}_\gamma = C_\gamma h_0^{-d\gamma}$, the total cost is:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{tol}^{-2} \left(\sum_{\mathrm{j}=1}^{\mathrm{L}} \sqrt{\tilde{\mathrm{C}}_\beta \tilde{\mathrm{C}}_\gamma \delta^{\mathrm{j}\frac{\beta - \mathrm{d}\gamma}{2}}} \right)^2 + \sum_{\mathrm{j}=0}^{\mathrm{L}} \mathrm{C}_\mathrm{j}$$

Proof.

We enforce the error constraint $\mathrm{MSE}(\hat{\mu}_{MLMC}) \leq \mathrm{tol}^2$ as

Bias constraint: $|\mathbb{E}[Q - Q_L]|^2 \leq \frac{1}{2}\mathrm{tol}^2$, Var. constraint: $\mathbb{V}\mathrm{ar}[\hat{\mu}_{\mathrm{MLMC}}] \leq \frac{1}{2}\mathrm{tol}^2$

From the Bias constraint we get

$$L(\mathrm{tol}) \equiv \mathrm{L} = \left\lceil \frac{\log(\sqrt{2}\mathrm{C}_\alpha \mathrm{h}_0^\alpha \mathrm{tol}^{-1})}{\alpha \log \delta^{-1}} \right\rceil \sim \log_\delta \mathrm{tol}^{\frac{1}{\alpha}} \ .$$

Setting $\tilde{C}_\beta = C_\beta h_0^\beta$ and $\tilde{C}_\gamma = C_\gamma h_0^{-d\gamma}$, the total cost is:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{tol}^{-2} \left(\sum_{\mathrm{j}=1}^{\mathrm{L}} \sqrt{\tilde{\mathrm{C}}_\beta \tilde{\mathrm{C}}_\gamma \delta^{\mathrm{j}\frac{\beta - \mathrm{d}\gamma}{2}}} \right)^2 + \sum_{\mathrm{j}=0}^{\mathrm{L}} \mathrm{C}_\mathrm{j}$$

Disregarding for the moment the term $\sum_j C_j$, we consider three cases:

- Case $\beta > d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{tol}^{-2}\left(\sum_{\mathrm{j}=1}^{\infty}\sqrt{\tilde{\mathrm{C}}_\beta\tilde{\mathrm{C}}_\gamma}\delta^{\mathrm{j}\frac{\beta-\mathrm{d}\gamma}{2}}\right)^2 \leq \mathrm{C}\,\mathrm{tol}^{-2}$$

- Case $\beta = d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{C}\,\mathrm{tol}^{-2}\mathrm{L} \leq \mathrm{C}\,\mathrm{tol}^{-2}(\log \mathrm{tol}^{-1})^2$$

- Case $\beta < d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{C}\mathrm{tol}^{-2}\delta^{\mathrm{L}\frac{\beta-\mathrm{d}\gamma}{2}} \leq \mathrm{C}\,\mathrm{tol}^{-2-\frac{\mathrm{d}\gamma-\beta}{\alpha}}$$

If, moreover, $2\alpha \geq \min(\beta, d\gamma)$ then the term $\sum_{j=0}^{L} C_j$ is of higher order that the terms above in all three cases. □

Exercise. Check that $\sum_{j=0}^{L} C_j$ is indeed of higher order for $2\alpha \geq \min(\beta, d\gamma)$ and under the assumptions above.

Disregarding for the moment the term $\sum_j C_j$, we consider three cases:

- Case $\beta > d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{tol}^{-2} \left(\sum_{j=1}^{\infty} \sqrt{\tilde{C}_\beta \tilde{C}_\gamma} \delta^{j \frac{\beta - d\gamma}{2}} \right)^2 \leq C\, \mathrm{tol}^{-2}$$

- Case $\beta = d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq C\, \mathrm{tol}^{-2} L \leq C\, \mathrm{tol}^{-2} (\log \mathrm{tol}^{-1})^2$$

- Case $\beta < d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq C \mathrm{tol}^{-2} \delta^{L \frac{\beta - d\gamma}{2}} \leq C\, \mathrm{tol}^{-2 - \frac{d\gamma - \beta}{\alpha}}$$

If, moreover, $2\alpha \geq \min(\beta, d\gamma)$ then the term $\sum_{j=0}^{L} C_j$ is of higher order that the terms above in all three cases. □

Exercise. Check that $\sum_{j=0}^{L} C_j$ is indeed of higher order for $2\alpha \geq \min(\beta, d\gamma)$ and under the assumptions above.

Disregarding for the moment the term $\sum_j C_j$, we consider three cases:

- Case $\beta > d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \le \mathrm{tol}^{-2} \left(\sum_{j=1}^{\infty} \sqrt{\tilde{C}_\beta \tilde{C}_\gamma} \delta^{j\frac{\beta - d\gamma}{2}} \right)^2 \le C\, \mathrm{tol}^{-2}$$

- Case $\beta = d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \le C\, \mathrm{tol}^{-2} L \le C\, \mathrm{tol}^{-2} (\log \mathrm{tol}^{-1})^2$$

- Case $\beta < d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \le C \mathrm{tol}^{-2} \delta^{L\frac{\beta - d\gamma}{2}} \le C\, \mathrm{tol}^{-2 - \frac{d\gamma - \beta}{\alpha}}$$

If, moreover, $2\alpha \ge \min(\beta, d\gamma)$ then the term $\sum_{j=0}^{L} C_j$ is of higher order that the terms above in all three cases. □

Exercise. Check that $\sum_{j=0}^{L} C_j$ is indeed of higher order for $2\alpha \ge \min(\beta, d\gamma)$ and under the assumptions above.

Disregarding for the moment the term $\sum_j C_j$, we consider three cases:

- Case $\beta > d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq \mathrm{tol}^{-2} \left(\sum_{j=1}^{\infty} \sqrt{\tilde{C}_\beta \tilde{C}_\gamma} \delta^{j \frac{\beta - d\gamma}{2}} \right)^2 \leq C \, \mathrm{tol}^{-2}$$

- Case $\beta = d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq C \, \mathrm{tol}^{-2} L \leq C \, \mathrm{tol}^{-2} (\log \mathrm{tol}^{-1})^2$$

- Case $\beta < d\gamma$:

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \leq C \mathrm{tol}^{-2} \delta^{L \frac{\beta - d\gamma}{2}} \leq C \, \mathrm{tol}^{-2 - \frac{d\gamma - \beta}{\alpha}}$$

If, moreover, $2\alpha \geq \min(\beta, d\gamma)$ then the term $\sum_{j=0}^{L} C_j$ is of higher order that the terms above in all three cases. □

Exercise. Check that $\sum_{j=0}^{L} C_j$ is indeed of higher order for $2\alpha \geq \min(\beta, d\gamma)$ and under the assumptions above.

Important (shocking ?) comments on the MLMC rates

Let us focus on two particular cases:

- Fast convergence rate, $\beta > d\gamma$.
  Here the complexity of MLMC is tol^{-2}, which is the same of Monte Carlo sampling when the cost to sample each realization is *fixed*. This means that we do not see the effect of the fine h_L discretization in the rates!
- Smooth noise, $\beta = 2\alpha$ and $\beta < d\gamma$.
  Here the resulting complexity is $\mathrm{tol}^{-\mathrm{d}\gamma/\alpha}$, which is the complexity of solving just **one** realization in the deepest level!

Further remarks:

- In all cases, MLMC has a better asymptotic complexity than MC. (in the pre-asymptotic regime, this is not always the case).
- The complexity analysis relies on the use of geometric meshes $h_\ell = h_0\delta^\ell$. Indeed, it can be shown that geometric refinement is nearly optimal [HajiAli-N.-vonSchwerin-Tempone 2016]

Important (shocking ?) comments on the MLMC rates

Let us focus on two particular cases:

- Fast convergence rate, $\beta > d\gamma$.
  Here the complexity of MLMC is tol^{-2}, which is the same of Monte Carlo sampling when the cost to sample each realization is *fixed*. This means that we do not see the effect of the fine h_L discretization in the rates!
- Smooth noise, $\beta = 2\alpha$ and $\beta < d\gamma$.
  Here the resulting complexity is $\mathrm{tol}^{-\mathrm{d}\gamma/\alpha}$, which is the complexity of solving just **one** realization in the deepest level!

Further remarks:

- In all cases, MLMC has a better asymptotic complexity than MC. (in the pre-asymptotic regime, this is not always the case).
- The complexity analysis relies on the use of geometric meshes $h_\ell = h_0\delta^\ell$. Indeed, it can be shown that geometric refinement is nearly optimal [HajiAli-N.-vonSchwerin-Tempone 2016]

Important (shocking ?) comments on the MLMC rates

Let us focus on two particular cases:

- Fast convergence rate, $\beta > d\gamma$.
  Here the complexity of MLMC is tol^{-2}, which is the same of Monte Carlo sampling when the cost to sample each realization is *fixed*. This means that we do not see the effect of the fine h_L discretization in the rates!

- Smooth noise, $\beta = 2\alpha$ and $\beta < d\gamma$.
  Here the resulting complexity is $\mathrm{tol}^{-\mathrm{d}\gamma/\alpha}$, which is the complexity of solving just **one** realization in the deepest level!

Further remarks:

- In all cases, MLMC has a better asymptotic complexity than MC. (in the pre-asymptotic regime, this is not always the case).
- The complexity analysis relies on the use of geometric meshes $h_\ell = h_0\delta^\ell$. Indeed, it can be shown that geometric refinement is nearly optimal [HajiAli-N.-vonSchwerin-Tempone 2016]

Important (shocking ?) comments on the MLMC rates

Let us focus on two particular cases:

- Fast convergence rate, $\beta > d\gamma$.
  Here the complexity of MLMC is tol^{-2}, which is the same of Monte Carlo sampling when the cost to sample each realization is *fixed*. This means that we do not see the effect of the fine h_L discretization in the rates!

- Smooth noise, $\beta = 2\alpha$ and $\beta < d\gamma$.
  Here the resulting complexity is $\mathrm{tol}^{-\mathrm{d}\gamma/\alpha}$, which is the complexity of solving just **one** realization in the deepest level!

Further remarks:

- In all cases, MLMC has a better asymptotic complexity than MC. (in the pre-asymptotic regime, this is not always the case).
- The complexity analysis relies on the use of geometric meshes $h_\ell = h_0\delta^\ell$. Indeed, it can be shown that geometric refinement is nearly optimal [HajiAli-N.-vonSchwerin-Tempone 2016]

Example 1 – stochastic differential equation

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t(y), \qquad t \in (0, T], \qquad X_0 = x_0$$

Quantity of interest: $Q = \tilde{Q}(X_T)$

discretized by Euler-Maruyama with step size $h_\ell = h_0 2^{-\ell}$. We have already seen that for smooth $a(\cdot)$ and $b(\cdot)$ one has

1. $|\mathbb{E}[Q - Q_\ell]| \lesssim h_\ell$ (order 1 convergence for the mean – weak rate)
2. $\mathbb{E}[(Q - Q_\ell)^2]^{\frac{1}{2}} \lesssim h_\ell^{\frac{1}{2}}$ (order 1/2 in mean square sense – strong rate)
3. $C_\ell \lesssim\lesssim h_\ell^{-1}$ (cost proportional to the numebr of iterations)

From 2 we deduce $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] \lesssim h_\ell$

Hence: $\alpha = 1$, $\beta = 1$, $d = 1$, $\gamma = 1$ $\implies$ $\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \simeq \mathrm{tol}^{-2} \log^2(\mathrm{tol})$

To be compared with $\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3}$

Example 1 – stochastic differential equation

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t(y), \qquad t \in (0, T], \qquad X_0 = x_0$$

Quantity of interest: $Q = \tilde{Q}(X_T)$

discretized by Euler-Maruyama with step size $h_\ell = h_0 2^{-\ell}$. We have already seen that for smooth $a(\cdot)$ and $b(\cdot)$ one has

1. $|\mathbb{E}[Q - Q_\ell]| \lesssim h_\ell$ (order 1 convergence for the mean – weak rate)
2. $\mathbb{E}[(Q - Q_\ell)^2]^{\frac{1}{2}} \lesssim h_\ell^{\frac{1}{2}}$ (order 1/2 in mean square sense – strong rate)
3. $C_\ell \lesssim\lesssim h_\ell^{-1}$ (cost proportional to the numebr of iterations)

From 2 we deduce $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] \lesssim h_\ell$

Hence: $\alpha = 1$, $\beta = 1$, $d = 1$, $\gamma = 1 \implies \mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \simeq \mathrm{tol}^{-2}\log^2(\mathrm{tol})$

To be compared with $\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3}$

Example 1 – stochastic differential equation

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t(y), \qquad t \in (0, T], \qquad X_0 = x_0$$

Quantity of interest: $\quad Q = \tilde{Q}(X_T)$

discretized by Euler-Maruyama with step size $h_\ell = h_0 2^{-\ell}$. We have already seen that for smooth $a(\cdot)$ and $b(\cdot)$ one has

1. $|\mathbb{E}[Q - Q_\ell]| \lesssim h_\ell$ (order 1 convergence for the mean – weak rate)
2. $\mathbb{E}[(Q - Q_\ell)^2]^{\frac{1}{2}} \lesssim h_\ell^{\frac{1}{2}}$ (order 1/2 in mean square sense – strong rate)
3. $C_\ell \lesssim\lesssim h_\ell^{-1}$ (cost proportional to the numebr of iterations)

From 2 we deduce $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] \lesssim h_\ell$

Hence: $\alpha = 1$, $\beta = 1$, $d = 1$, $\gamma = 1 \quad \Longrightarrow \quad \mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \simeq \mathrm{tol}^{-2} \log^2(\mathrm{tol})$

To be compared with $\mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3}$

Example 2 – PDE with random parameters

$$-\operatorname{div}(a(y)\nabla u) = f, \quad \text{in } D \subset \mathbb{R}^d, \qquad u = 0, \text{ on } \partial D$$

and $Q = \tilde{Q}(u)$ a Lipschitz functional. Discretization by $\mathbb{P}^1$ finite elements on a regular triangulation with mesh size h. Under suitable assumptions

1. $|Q(y) - Q_\ell(y)| \leq C h_\ell, \ \forall y \in \Gamma$ (order 1 strong convergence)
2. $C_\ell \lesssim h_\ell^{-d\gamma}$

From 1. we infer $|\mathbb{E}[Q - Q_\ell]| \lesssim h_\ell$ and $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] \lesssim h_\ell^2$.

2D case: $\alpha = 1$, $\beta = 2$, $d = 2$, and $\gamma = 1$ (optimal solver) ($\beta = d\gamma$)

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \simeq \mathrm{tol}^{-2} \log^2(\mathrm{tol}) \qquad \| \qquad \mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-4}$$

3D case: $\alpha = 1$, $\beta = 2$, $d = 3$, and $\gamma = 1$ (optimal solver) ($\beta < d\gamma$)

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3} \qquad \| \qquad \mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-5}$$

Example 2 – PDE with random parameters

$$-\operatorname{div}(a(y)\nabla u) = f, \quad \text{in } D \subset \mathbb{R}^d, \qquad u = 0, \ \text{on } \partial D$$

and $Q = \tilde{Q}(u)$ a Lipschitz functional. Discretization by $\mathbb{P}^1$ finite elements on a regular triangulation with mesh size h. Under suitable assumptions

1. $|Q(y) - Q_\ell(y)| \leq C h_\ell, \ \forall y \in \Gamma$ (order 1 strong convergence)
2. $C_\ell \lesssim h_\ell^{-d\gamma}$

From 1. we infer $|\mathbb{E}[Q - Q_\ell]| \lesssim h_\ell$ and $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] \lesssim h_\ell^2$.

2D case: $\alpha = 1$, $\beta = 2$, $d = 2$, and $\gamma = 1$ (optimal solver) ($\beta = d\gamma$)

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \simeq \mathrm{tol}^{-2} \log^2(\mathrm{tol}) \qquad \| \qquad \mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-4}$$

3D case: $\alpha = 1$, $\beta = 2$, $d = 3$, and $\gamma = 1$ (optimal solver) ($\beta < d\gamma$)

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3} \qquad \| \qquad \mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-5}$$

Example 2 – PDE with random parameters

$$-\operatorname{div}(a(y)\nabla u) = f, \quad \text{in } D \subset \mathbb{R}^d, \qquad u = 0, \ \text{on } \partial D$$

and $Q = \tilde{Q}(u)$ a Lipschitz functional. Discretization by $\mathbb{P}^1$ finite elements on a regular triangulation with mesh size h. Under suitable assumptions

1. $|Q(y) - Q_\ell(y)| \leq Ch_\ell, \ \forall y \in \Gamma$ (order 1 strong convergence)
2. $C_\ell \lesssim h_\ell^{-d\gamma}$

From 1. we infer $|\mathbb{E}[Q - Q_\ell]| \lesssim h_\ell$ and $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] \lesssim h_\ell^2$.

2D case: $\alpha = 1$, $\beta = 2$, $d = 2$, and $\gamma = 1$ (optimal solver) ($\beta = d\gamma$)

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \simeq \mathrm{tol}^{-2}\log^2(\mathrm{tol}) \qquad \| \qquad \mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-4}$$

3D case: $\alpha = 1$, $\beta = 2$, $d = 3$, and $\gamma = 1$ (optimal solver) ($\beta < d\gamma$)

$$\mathrm{Cost}(\hat{\mu}_{MLMC}, \mathrm{tol}) \simeq \mathrm{tol}^{-3} \qquad \| \qquad \mathrm{Cost}(\hat{\mu}_{MC}, \mathrm{tol}) \simeq \mathrm{tol}^{-5}$$

Controlling the error in MLMC

Recall: $\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}$

Given a hierarchy $\{M_\ell\}_\ell$ and samples $\{\Delta_\ell Q(y^{(i,\ell)}),\ i = 1, \dots, M_\ell\}_{\ell=0}^{L}$, with $\Delta_\ell Q = Q_\ell - Q_{\ell-1}$

Bias estimation (as in MC): use a posteriori error estimators or extrapolation strategies. E.g. Richardson extrapolation

$$B \approx \hat{B}_L := \frac{\hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]}{\delta^{-\alpha} - 1}$$

where the weak rate α is either known a priori or estimated from $\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}]$, $\ell = 1, \dots, L$.

Variance estimation: use sample variance estimator

$$\hat{V}_{MLMC} := \sum_{\ell=0}^{L} \frac{\hat{V}_\ell}{M_\ell} \quad \text{with} \quad \hat{V}_\ell = \widehat{\mathrm{Var}}_{M_\ell}[Q_\ell - Q_{\ell-1}]$$

Cost estimation: $\hat{C}_\ell := \hat{\mathbb{E}}_{M_\ell}[\mathrm{Cost}(Q_\ell - Q_{\ell-1})]$.

Controlling the error in MLMC

Recall: $\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}$

Given a hierarchy $\{M_\ell\}_\ell$ and samples $\{\Delta_\ell Q(y^{(i,\ell)}),\ i = 1, \ldots, M_\ell\}_{\ell=0}^{L}$, with $\Delta_\ell Q = Q_\ell - Q_{\ell-1}$

Bias estimation (as in MC): use a posteriori error estimators or extrapolation strategies. E.g. Richardson extrapolation

$$B \approx \hat{B}_L := \frac{\hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]}{\delta^{-\alpha} - 1}$$

where the weak rate α is either known a priori or estimated from $\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}],\ \ell = 1, \ldots, L$.

Variance estimation: use sample variance estimator

$$\hat{V}_{MLMC} := \sum_{\ell=0}^{L} \frac{\hat{V}_\ell}{M_\ell} \quad \text{with} \quad \hat{V}_\ell = \widehat{\mathrm{Var}}_{M_\ell}[Q_\ell - Q_{\ell-1}]$$

Cost estimation: $\hat{C}_\ell := \hat{\mathbb{E}}_{M_\ell}[\mathrm{Cost}(Q_\ell - Q_{\ell-1})]$.

Controlling the error in MLMC

Recall: $\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}$

Given a hierarchy $\{M_\ell\}_\ell$ and samples $\{\Delta_\ell Q(y^{(i,\ell)}),\ i = 1, \dots, M_\ell\}_{\ell=0}^{L}$, with $\Delta_\ell Q = Q_\ell - Q_{\ell-1}$

Bias estimation (as in MC): use a posteriori error estimators or extrapolation strategies. E.g. Richardson extrapolation

$$B \approx \hat{B}_L := \frac{\hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]}{\delta^{-\alpha} - 1}$$

where the weak rate α is either known a priori or estimated from $\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}],\ \ell = 1, \dots, L$.

Variance estimation: use sample variance estimator

$$\hat{V}_{MLMC} := \sum_{\ell=0}^{L} \frac{\hat{V}_\ell}{M_\ell} \quad \text{with} \quad \hat{V}_\ell = \widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[Q_\ell - Q_{\ell-1}]$$

Cost estimation: $\hat{C}_\ell := \hat{\mathbb{E}}_{M_\ell}[\mathrm{Cost}(Q_\ell - Q_{\ell-1})]$.

Controlling the error in MLMC

Recall: $\mathrm{MSE}(\hat{\mu}_{MLMC}) = B^2 + V_{MLMC} = \mathbb{E}[Q - Q_L]^2 + \sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}$

Given a hierarchy $\{M_\ell\}_\ell$ and samples $\{\Delta_\ell Q(y^{(i,\ell)}),\ i = 1, \dots, M_\ell\}_{\ell=0}^{L}$, with $\Delta_\ell Q = Q_\ell - Q_{\ell-1}$

Bias estimation (as in MC): use a posteriori error estimators or extrapolation strategies. E.g. Richardson extrapolation

$$B \approx \hat{B}_L := \frac{\hat{\mathbb{E}}_{M_L}[Q_L - Q_{L-1}]}{\delta^{-\alpha} - 1}$$

where the weak rate α is either known a priori or estimated from $\hat{\mathbb{E}}_{M_\ell}[Q_\ell - Q_{\ell-1}],\ \ell = 1, \dots, L$.

Variance estimation: use sample variance estimator

$$\hat{V}_{MLMC} := \sum_{\ell=0}^{L} \frac{\hat{V}_\ell}{M_\ell} \quad \text{with} \quad \hat{V}_\ell = \widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[Q_\ell - Q_{\ell-1}]$$

Cost estimation: $\hat{C}_\ell := \hat{\mathbb{E}}_{M_\ell}[\mathrm{Cost}(Q_\ell - Q_{\ell-1})]$.

Adaptive MLMC

Error splitting we aim at $B^2 \leq \frac{\mathrm{tol}^2}{2}$ and $V \leq \frac{\mathrm{tol}^2}{2}$

Given a MLMC run and estimates $\hat{B}_L$ and $\hat{V}_{MLMC}$

- if $|\hat{B}_L| > \frac{tol}{\sqrt{2}}$ $\Longrightarrow$ set $L = L + 1$ and run $\bar{M}$ simulations to estimate $\hat{V}_L$
- it $\hat{V}_{MLMC} > \frac{tol^2}{2}$ $\Longrightarrow$ compute optimal $\{M_\ell\}_\ell$ using the formula

$$M_\ell = \left\lceil \frac{2}{\mathrm{tol}^2} \sqrt{\frac{\hat{V}_\ell}{\hat{C}_\ell}} \sum_{j=0}^{L} \sqrt{\hat{V}_\ell \hat{C}_\ell} \right\rceil$$

and run the extra simulations needed

Adaptive MLMC

Error splitting we aim at $B^2 \leq \frac{\mathrm{tol}^2}{2}$ and $V \leq \frac{\mathrm{tol}^2}{2}$

Given a MLMC run and estimates $\hat{B}_L$ and $\hat{V}_{MLMC}$

- if $|\hat{B}_L| > \frac{tol}{\sqrt{2}}$ $\implies$ set $L = L+1$ and run $\bar{M}$ simulations to estimate $\hat{V}_L$
- it $\hat{V}_{MLMC} > \frac{tol^2}{2}$ $\implies$ compute optimal $\{M_\ell\}_\ell$ using the formula

$$M_\ell = \left\lceil \frac{2}{\mathrm{tol}^2} \sqrt{\frac{\hat{V}_\ell}{\hat{C}_\ell}} \sum_{j=0}^{L} \sqrt{\hat{V}_\ell \hat{C}_\ell} \right\rceil$$

and run the extra simulations needed

A simple adaptive MLMC algorithm

Algorithm (Adaptive MLMC, from [Giles Acta Num. 2015])

1. start with $L = 2$, and initial $M_0 = M_1 = M_2 = \bar{M}$ on levels $\ell = 0, 1, 2$
2. while extra samples need to be evaluated do
   - 2.1 evaluate extra samples on each level
   - 2.2 compute/update estimates $\hat{V}_\ell$, $\ell = 0, \ldots, L$
   - 2.3 define optimal M_ℓ, $\ell = 0, \ldots, L$
   - 2.4 if $\hat{B}_L > \frac{\mathrm{tol}}{\sqrt{2}}$ set $L := L + 1$, and initialise $M_L = \bar{M}$
3. end while

Drawback of the simple algorithm

- The initialization $M_L = \bar{M}$ on finest level L may be too costly (in the best scenario only a couple of simulations are needed on level L)
- The sample variance estimator $\hat{V}_\ell = \widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[Q_\ell - Q_{\ell-1}]$ may be unreliable for M_ℓ small, which typically happens in finest levels.

Estimation of V_ℓ on finest levels need to be combined with suitable extrapolation from previous levels.

E.g. [Giles 2015] proposes $\hat{V}_\ell = \max\{\widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[\Delta_\ell Q], \frac{1}{2}\delta^\beta V_{\ell-1}\}$.

A simple adaptive MLMC algorithm

Algorithm (Adaptive MLMC, from [Giles Acta Num. 2015])

1. start with $L = 2$, and initial $M_0 = M_1 = M_2 = \bar{M}$ on levels $\ell = 0, 1, 2$
2. while extra samples need to be evaluated do
   - 2.1 evaluate extra samples on each level
   - 2.2 compute/update estimates $\hat{V}_\ell$, $\ell = 0, \dots, L$
   - 2.3 define optimal M_ℓ, $\ell = 0, \dots, L$
   - 2.4 if $\hat{B}_L > \frac{\mathrm{tol}}{\sqrt{2}}$ set $L := L + 1$, and initialise $M_L = \bar{M}$
3. end while

Drawback of the simple algorithm

- The initialization $M_L = \bar{M}$ on finest level L may be too costly (in the best scenario only a couple of simulations are needed on level L)
- The sample variance estimator $\hat{V}_\ell = \widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[Q_\ell - Q_{\ell-1}]$ may be unreliable for M_ℓ small, which typically happens in finest levels.

Estimation of V_ℓ on finest levels need to be combined with suitable extrapolation from previous levels.

E.g. [Giles 2015] proposes $\hat{V}_\ell = \max\{\widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[\Delta_\ell Q], \frac{1}{2}\delta^\beta V_{\ell-1}\}$.

A simple adaptive MLMC algorithm

Algorithm (Adaptive MLMC, from [Giles Acta Num. 2015])

1. start with $L = 2$, and initial $M_0 = M_1 = M_2 = \bar{M}$ on levels $\ell = 0, 1, 2$
2. while extra samples need to be evaluated do
   - 2.1 evaluate extra samples on each level
   - 2.2 compute/update estimates $\hat{V}_\ell$, $\ell = 0, \dots, L$
   - 2.3 define optimal M_ℓ, $\ell = 0, \dots, L$
   - 2.4 if $\hat{B}_L > \frac{\mathrm{tol}}{\sqrt{2}}$ set $L := L + 1$, and initialise $M_L = \bar{M}$
3. end while

Drawback of the simple algorithm

- The initialization $M_L = \bar{M}$ on finest level L may be too costly (in the best scenario only a couple of simulations are needed on level L)
- The sample variance estimator $\hat{V}_\ell = \widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[Q_\ell - Q_{\ell-1}]$ may be unreliable for M_ℓ small, which typically happens in finest levels.

Estimation of V_ℓ on finest levels need to be combined with suitable extrapolation from previous levels.

E.g. [Giles 2015] proposes $\hat{V}_\ell = \max\{\widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[\Delta_\ell Q], \frac{1}{2}\delta^\beta V_{\ell-1}\}$.

A simple adaptive MLMC algorithm

Algorithm (Adaptive MLMC, from [Giles Acta Num. 2015])

1. start with $L = 2$, and initial $M_0 = M_1 = M_2 = \bar{M}$ on levels $\ell = 0, 1, 2$
2. while extra samples need to be evaluated do
   - 2.1 evaluate extra samples on each level
   - 2.2 compute/update estimates $\hat{V}_\ell$, $\ell = 0, \dots, L$
   - 2.3 define optimal M_ℓ, $\ell = 0, \dots, L$
   - 2.4 if $\hat{B}_L > \frac{\mathrm{tol}}{\sqrt{2}}$ set $L := L + 1$, and initialise $M_L = \bar{M}$
3. end while

Drawback of the simple algorithm

- The initialization $M_L = \bar{M}$ on finest level L may be too costly (in the best scenario only a couple of simulations are needed on level L)
- The sample variance estimator $\hat{V}_\ell = \widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[Q_\ell - Q_{\ell-1}]$ may be unreliable for M_ℓ small, which typically happens in finest levels.

Estimation of V_ℓ on finest levels need to be combined with suitable extrapolation from previous levels.

E.g. [Giles 2015] proposes $\hat{V}_\ell = \max\{\widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[\Delta_\ell Q], \frac{1}{2}\delta^\beta V_{\ell-1}\}$.

Continuation Multilevel Monte Carlo (CMLMC)

[Collier-HajiAli-N.-vonSchwerin-Tempone 2015, Pisaroni-N.-Leyland 2017]

Idea: Solve the problem with decreasing tolerances $tol^{(0)} > tol^{(1)} > \ldots \geq tol$. Use collected samples on all levels to improve the estimate of $V_\ell = \mathbb{V}\mathrm{ar}[\Delta_\ell Q]$ and $\mu_\ell = \mathbb{E}[Q - Q_\ell]$.

MAP Bayesian estimator $\hat{V}_\ell$ at iteration j:

- we make the ansatz $\Delta_\ell Q \sim N(\mu_\ell, V_\ell)$
- based on acquired samples at previous iteration, we fit models (least squares)
  - $\mu_\ell^{model} = C_\alpha h_\ell^\alpha$
  - $V_\ell^{model} = C_\beta h_\ell^\beta$
- We take a Normal-Gamma prior for (μ_ℓ, V_ℓ), with mode in $(\mu_\ell^{model}, V_\ell^{model})$
- Then $\hat{V}_\ell$ is the MAP Bayesian estimator based on the Normal-Gamma prior and the actual samples acquired at iteration j

Effectively, we have

$$M_\ell = 0 \qquad \hat{V}_\ell = V_\ell^{model} \qquad \text{(prior model)}$$

$$M_\ell \to \infty \qquad \hat{V}_\ell \approx \widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[\Delta_\ell Q] \qquad \text{(sample variance)}$$

$\hat{V}_\ell$ is then used to determine the sample sizes M_ℓ for the next iteration.

Continuation Multilevel Monte Carlo (CMLMC)

[Collier-HajiAli-N.-vonSchwerin-Tempone 2015, Pisaroni-N.-Leyland 2017]

Idea: Solve the problem with decreasing tolerances $tol^{(0)} > tol^{(1)} > \ldots \geq tol$.
Use collected samples on all levels to improve the estimate of $V_\ell = \mathbb{V}\mathrm{ar}[\Delta_\ell Q]$ and $\mu_\ell = \mathbb{E}[Q - Q_\ell]$.

MAP Bayesian estimator $\hat{V}_\ell$ at iteration j:

- we make the ansatz $\Delta_\ell Q \sim N(\mu_\ell, V_\ell)$
- based on acquired samples at previous iteration, we fit models (least squares)
  - $\mu_\ell^{model} = C_\alpha h_\ell^\alpha$
  - $V_\ell^{model} = C_\beta h_\ell^\beta$
- We take a Normal-Gamma prior for (μ_ℓ, V_ℓ), with mode in $(\mu_\ell^{model}, V_\ell^{model})$
- Then $\hat{V}_\ell$ is the MAP Bayesian estimator based on the Normal-Gamma prior and the actual samples acquired at iteration j

Effectively, we have

$M_\ell = 0$	$\hat{V}_\ell = V_\ell^{model}$	(prior model)
$M_\ell \to \infty$	$\hat{V}_\ell \approx \widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[\Delta_\ell Q]$	(sample variance)

$\hat{V}_\ell$ is then used to determine the sample sizes M_ℓ for the next iteration.

Continuation Multilevel Monte Carlo (CMLMC)

[Collier-HajiAli-N.-vonSchwerin-Tempone 2015, Pisaroni-N.-Leyland 2017]

Idea: Solve the problem with decreasing tolerances $tol^{(0)} > tol^{(1)} > \ldots \geq tol$.
Use collected samples on all levels to improve the estimate of $V_\ell = \mathbb{V}\mathrm{ar}[\Delta_\ell Q]$ and $\mu_\ell = \mathbb{E}[Q - Q_\ell]$.

MAP Bayesian estimator $\hat{V}_\ell$ at iteration j:

- we make the ansatz $\Delta_\ell Q \sim N(\mu_\ell, V_\ell)$
- based on acquired samples at previous iteration, we fit models (least squares)
  - $\mu_\ell^{model} = C_\alpha h_\ell^\alpha$
  - $V_\ell^{model} = C_\beta h_\ell^\beta$
- We take a Normal-Gamma prior for (μ_ℓ, V_ℓ), with mode in $(\mu_\ell^{model}, V_\ell^{model})$
- Then $\hat{V}_\ell$ is the MAP Bayesian estimator based on the Normal-Gamma prior and the actual samples acquired at iteration j

Effectively, we have

$$M_\ell = 0 \qquad \hat{V}_\ell = V_\ell^{model} \qquad \text{(prior model)}$$

$$M_\ell \to \infty \qquad \hat{V}_\ell \approx \widehat{\mathbb{V}\mathrm{ar}}_{M_\ell}[\Delta_\ell Q] \qquad \text{(sample variance)}$$

$\hat{V}_\ell$ is then used to determine the sample sizes M_ℓ for the next iteration.

The CMLMC algorithm

Choose a sequence of decreasing tolerances: $\mathrm{tol}_0 > \mathrm{tol}_1 > \ldots > \mathrm{tol}_K = \mathrm{tol}$ and an initial guess of the rates $(\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)})$, constants $(C_\alpha^{(0)}, C_\beta^{(0)}, C_\gamma^{(0)})$ and variances $\{\hat{V}_\ell^{(0)}\}_{\ell=0}^{L^{(0)}}$,

for $k = 1, \ldots, K$

Based on rates $(\alpha^{(k-1)}, \beta^{(k-1)}, \gamma^{(k-1)})$, constants $(C_\alpha^{(k-1)}, C_\beta^{(k-1)}, C_\gamma^{(k-1)})$ and variances $\{\hat{V}_\ell^{(k-1)}\}_{\ell=0}^{L^{(k-1)}}$

- compute optimal $L^{(k)}$ s.t. $C_\alpha^{(k-1)} h_L^{\alpha^{(k-1)}} \leq \frac{\mathrm{tol}_k}{2}$
- compute optimal $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ s.t. $\sum_{\ell=0}^{L^{(k)}} \frac{\hat{V}_\ell^{(k-1)}}{M_\ell} \leq \frac{\mathrm{tol}_k^2}{2}$.
- run **MLMC** with $L^{(k)}$, $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$
- **update** rates $(\alpha^{(k)}, \beta^{(k)}, \gamma^{(k)})$, constants $(C_\alpha^{(k)}, C_\beta^{(k)}, C_\gamma^{(k)})$ and variances $\{\hat{V}_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ based on the new simulations performed
- $k = k + 1$

end for

The CMLMC algorithm

Choose a sequence of decreasing tolerances: $\mathrm{tol}_0 > \mathrm{tol}_1 > \ldots > \mathrm{tol}_K = \mathrm{tol}$ and an initial guess of the rates $(\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)})$, constants $(C_\alpha^{(0)}, C_\beta^{(0)}, C_\gamma^{(0)})$ and variances $\{\hat{V}_\ell^{(0)}\}_{\ell=0}^{L^{(0)}}$,

for $k = 1, \ldots, K$

Based on rates $(\alpha^{(k-1)}, \beta^{(k-1)}, \gamma^{(k-1)})$, constants $(C_\alpha^{(k-1)}, C_\beta^{(k-1)}, C_\gamma^{(k-1)})$ and variances $\{\hat{V}_\ell^{(k-1)}\}_{\ell=0}^{L^{(k-1)}}$

- compute optimal $L^{(k)}$ s.t. $C_\alpha^{(k-1)} h_L^{\alpha^{(k-1)}} \leq \frac{\mathrm{tol}_k}{2}$
- compute optimal $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ s.t. $\sum_{\ell=0}^{L^{(k)}} \frac{\hat{V}_\ell^{(k-1)}}{M_\ell} \leq \frac{\mathrm{tol}_k^2}{2}$.
- run **MLMC** with $L^{(k)}$, $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$
- **update** rates $(\alpha^{(k)}, \beta^{(k)}, \gamma^{(k)})$, constants $(C_\alpha^{(k)}, C_\beta^{(k)}, C_\gamma^{(k)})$ and variances $\{\hat{V}_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ based on the new simulations performed
- $k = k + 1$

end for

The CMLMC algorithm

Choose a sequence of decreasing tolerances: $\mathrm{tol}_0 > \mathrm{tol}_1 > \ldots > \mathrm{tol}_K = \mathrm{tol}$ and an initial guess of the rates $(\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)})$, constants $(C_\alpha^{(0)}, C_\beta^{(0)}, C_\gamma^{(0)})$ and variances $\{\hat{V}_\ell^{(0)}\}_{\ell=0}^{L^{(0)}}$,

for $k = 1, \ldots, K$

Based on rates $(\alpha^{(k-1)}, \beta^{(k-1)}, \gamma^{(k-1)})$, constants $(C_\alpha^{(k-1)}, C_\beta^{(k-1)}, C_\gamma^{(k-1)})$ and variances $\{\hat{V}_\ell^{(k-1)}\}_{\ell=0}^{L^{(k-1)}}$

- compute optimal $L^{(k)}$ s.t. $C_\alpha^{(k-1)} h_L^{\alpha^{(k-1)}} \leq \frac{\mathrm{tol}_k}{2}$
- compute optimal $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ s.t. $\sum_{\ell=0}^{L^{(k)}} \frac{\hat{V}_\ell^{(k-1)}}{M_\ell} \leq \frac{\mathrm{tol}_k^2}{2}$.
- run **MLMC** with $L^{(k)}$, $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$
- **update** rates $(\alpha^{(k)}, \beta^{(k)}, \gamma^{(k)})$, constants $(C_\alpha^{(k)}, C_\beta^{(k)}, C_\gamma^{(k)})$ and variances $\{\hat{V}_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ based on the new simulations performed
- $k = k + 1$

end for

The CMLMC algorithm

Choose a sequence of decreasing tolerances: $\mathrm{tol}_0 > \mathrm{tol}_1 > \ldots > \mathrm{tol}_K = \mathrm{tol}$ and an initial guess of the rates $(\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)})$, constants $(C_\alpha^{(0)}, C_\beta^{(0)}, C_\gamma^{(0)})$ and variances $\{\hat{V}_\ell^{(0)}\}_{\ell=0}^{L^{(0)}}$,

for $k = 1, \ldots, K$

Based on rates $(\alpha^{(k-1)}, \beta^{(k-1)}, \gamma^{(k-1)})$, constants $(C_\alpha^{(k-1)}, C_\beta^{(k-1)}, C_\gamma^{(k-1)})$ and variances $\{\hat{V}_\ell^{(k-1)}\}_{\ell=0}^{L^{(k-1)}}$

- compute optimal $L^{(k)}$ s.t. $C_\alpha^{(k-1)} h_L^{\alpha^{(k-1)}} \leq \frac{\mathrm{tol}_k}{2}$
- compute optimal $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ s.t. $\sum_{\ell=0}^{L^{(k)}} \frac{\hat{V}_\ell^{(k-1)}}{M_\ell} \leq \frac{\mathrm{tol}_k^2}{2}$.
- run **MLMC** with $L^{(k)}$, $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$
- **update** rates $(\alpha^{(k)}, \beta^{(k)}, \gamma^{(k)})$, constants $(C_\alpha^{(k)}, C_\beta^{(k)}, C_\gamma^{(k)})$ and variances $\{\hat{V}_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ based on the new simulations performed
- $k = k + 1$

end for

The CMLMC algorithm

Choose a sequence of decreasing tolerances: $\mathrm{tol}_0 > \mathrm{tol}_1 > \ldots > \mathrm{tol}_\mathrm{K} = \mathrm{tol}$ and an initial guess of the rates $(\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)})$, constants $(C_\alpha^{(0)}, C_\beta^{(0)}, C_\gamma^{(0)})$ and variances $\{\hat{V}_\ell^{(0)}\}_{\ell=0}^{L^{(0)}}$,

for $k = 1, \ldots, K$

Based on rates $(\alpha^{(k-1)}, \beta^{(k-1)}, \gamma^{(k-1)})$, constants $(C_\alpha^{(k-1)}, C_\beta^{(k-1)}, C_\gamma^{(k-1)})$ and variances $\{\hat{V}_\ell^{(k-1)}\}_{\ell=0}^{L^{(k-1)}}$

- compute optimal $L^{(k)}$ s.t. $C_\alpha^{(k-1)} h_L^{\alpha^{(k-1)}} \leq \frac{\mathrm{tol_k}}{2}$
- compute optimal $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ s.t. $\sum_{\ell=0}^{L^{(k)}} \frac{\hat{V}_\ell^{(k-1)}}{M_\ell} \leq \frac{\mathrm{tol_k^2}}{2}$.
- run **MLMC** with $L^{(k)}$, $\{M_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$
- **update** rates $(\alpha^{(k)}, \beta^{(k)}, \gamma^{(k)})$, constants $(C_\alpha^{(k)}, C_\beta^{(k)}, C_\gamma^{(k)})$ and variances $\{\hat{V}_\ell^{(k)}\}_{\ell=0}^{L^{(k)}}$ based on the new simulations performed
- $k = k + 1$

end for

Alternative error splitting based on CLT

It has been shown in [Collier-HajiAli-N.-vonSchwerin-Tempone, 2015], [Hoel-Krumscheid, 2019] that the estimator $\hat{\mu}_{MLMC}$ satisfies a CLT. More precisely, taking $L = L(\mathrm{tol})$ to satisfy a bias condition and $M_\ell = M_\ell(\mathrm{tol})$ with optimal allocation to satisfy the variance condition, under mild assumptions

$$\frac{\hat{\mu}_{MLMC} - \mathbb{E}[\hat{\mu}_{MLMC}]}{\sqrt{\mathbb{V}\mathrm{ar}[\hat{\mu}_{MLMC}]}} \xrightarrow{d} N(0,1)$$

Alternative Error splitting for aymptotic confidence level $1-\delta$

$$|\hat{B}_L| \approx (1-\theta)\mathrm{tol}, \qquad \mathrm{c}_\delta \sqrt{\sum_{\ell=0}^{\mathrm{L}} \frac{\hat{\mathrm{V}}_\ell}{\mathrm{M}_\ell}} \approx \theta \mathrm{tol}$$

CMLMC can also estimate the optimal splitting parameter: at iteration k

$$(L^{(k)}, \theta^{(k)}) = \underset{\substack{\theta\in(0,1)\\ L^{(k-1)} \le L \le L_{max}}}{\mathrm{argmin}} \mathrm{Cost}^{(k-1)}(L,\theta), \qquad \text{s.t. } C_\alpha^{(k-1)} h_L^{\alpha^{(k-1)}} \le (1-\theta)\mathrm{tol}_\mathrm{k}$$

Alternative error splitting based on CLT

It has been shown in [Collier-HajiAli-N.-vonSchwerin-Tempone, 2015], [Hoel-Krumscheid, 2019] that the estimator $\hat{\mu}_{MLMC}$ satisfies a CLT. More precisely, taking $L = L(\mathrm{tol})$ to satisfy a bias condition and $M_\ell = M_\ell(\mathrm{tol})$ with optimal allocation to satisfy the variance condition, under mild assumptions

$$\frac{\hat{\mu}_{MLMC} - \mathbb{E}[\hat{\mu}_{MLMC}]}{\sqrt{\mathbb{V}\mathrm{ar}[\hat{\mu}_{MLMC}]}} \xrightarrow{d} N(0,1)$$

Alternative Error splitting for aymptotic confidence level $1-\delta$

$$|\hat{B}_L| \approx (1-\theta)\mathrm{tol}, \qquad \mathrm{c}_\delta \sqrt{\sum_{\ell=0}^{\mathrm{L}} \frac{\hat{\mathrm{V}}_\ell}{\mathrm{M}_\ell}} \approx \theta \mathrm{tol}$$

CMLMC can also estimate the optimal splitting parameter: at iteration k

$$(L^{(k)}, \theta^{(k)}) = \underset{\substack{\theta\in(0,1) \\ L^{(k-1)}\leq L\leq L_{max}}}{\mathrm{argmin}} \mathrm{Cost}^{(k-1)}(L,\theta), \qquad \text{s.t. } C_\alpha^{(k-1)} h_L^{\alpha^{(k-1)}} \leq (1-\theta)\mathrm{tol_k}$$

Alternative error splitting based on CLT

It has been shown in [Collier-HajiAli-N.-vonSchwerin-Tempone, 2015], [Hoel-Krumscheid, 2019] that the estimator $\hat{\mu}_{MLMC}$ satisfies a CLT. More precisely, taking $L = L(\mathrm{tol})$ to satisfy a bias condition and $M_\ell = M_\ell(\mathrm{tol})$ with optimal allocation to satisfy the variance condition, under mild assumptions

$$\frac{\hat{\mu}_{MLMC} - \mathbb{E}[\hat{\mu}_{MLMC}]}{\sqrt{\mathbb{V}\mathrm{ar}[\hat{\mu}_{MLMC}]}} \xrightarrow{d} N(0,1)$$

Alternative Error splitting for aymptotic confidence level $1-\delta$

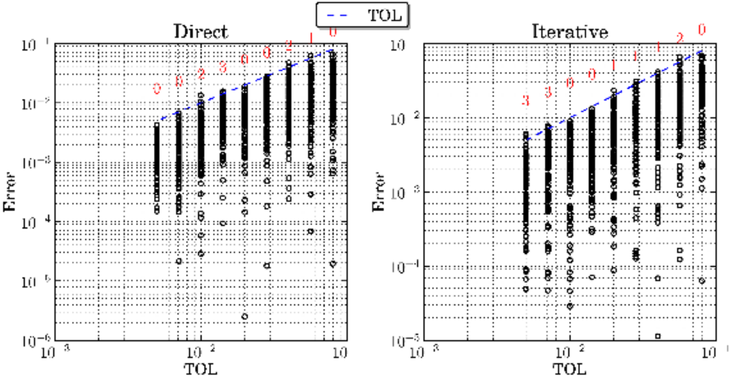
$$|\hat{B}_L| \approx (1-\theta)\mathrm{tol}, \qquad \mathrm{c}_\delta \sqrt{\sum_{\ell=0}^{\mathrm{L}} \frac{\hat{\mathrm{V}}_\ell}{\mathrm{M}_\ell}} \approx \theta \mathrm{tol}$$

CMLMC can also estimate the optimal splitting parameter: at iteration k

$$(L^{(k)}, \theta^{(k)}) = \underset{\substack{\theta \in (0,1) \\ L^{(k-1)} \leq L \leq L_{max}}}{\operatorname{argmin}} \ \mathrm{Cost}^{(k-1)}(L, \theta), \qquad \text{s.t. } C_\alpha^{(k-1)} h_L^{\alpha^{(k-1)}} \leq (1-\theta)\mathrm{tol_k}$$

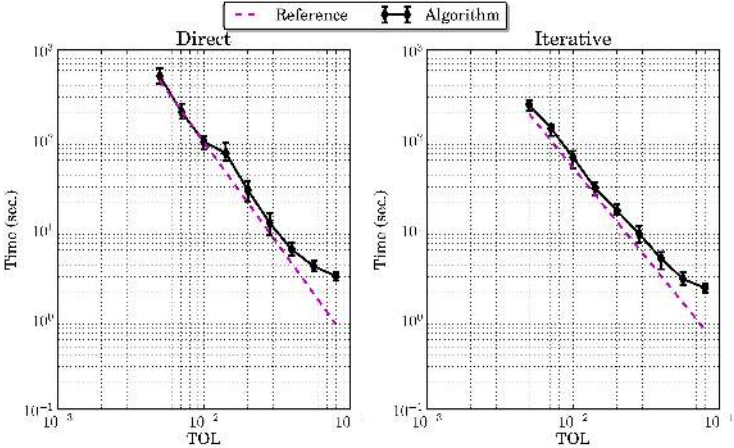
Error plot of CMLMC

3D elliptic PDE with random coefficients; $\mathbb{P}_1$ finite elements, smooth functional: $\alpha = 2$, $\beta = 4$, $\gamma = 1$ (iterative), $\gamma = 1.5$ (sparse direct)
Error splitting based on CLT with condfidence $1 - \delta$. Exact solution is known so true error can be measured and compared with prescribed tolerance.

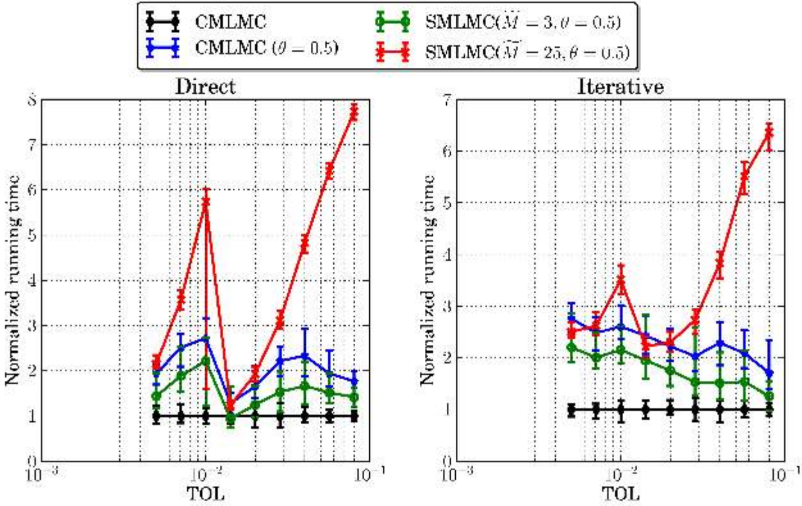


The algorithm was run with $c_\delta = 2$ so that the bound holds with 95% confidence.

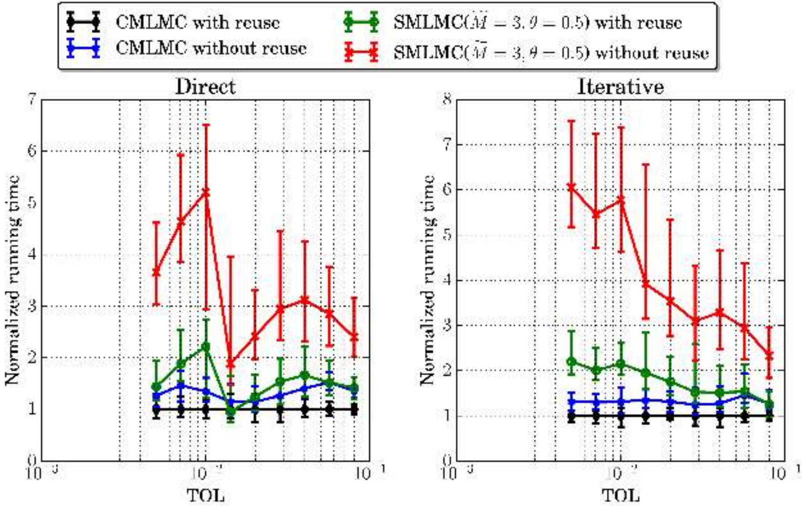
Total work of CMLMC



Reference lines are $\text{tol}^{-2.25}$ and tol^{-2}, respectively. This is consistent with complexity analysisx.



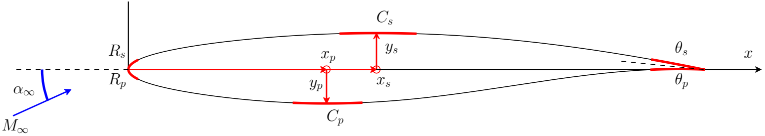
Improvement in running time due to better choice of splitting parameter, θ.



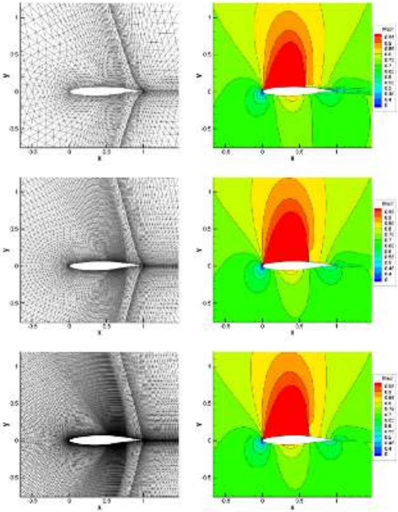
Reusing samples in CMLMC does not significantly improve running, since the work is dominated by the work of the last iteration.

Computation of C_L and pressure coeff. for RAE2822 airfoil

	Parameter	Reference value (r)	Uncertainty
Operational	α_∞	2.31°	$\mathcal{TN}(r, 2\%r, 90\%r, 100\%r)$
	M_∞	0.729	$\mathcal{TN}(r, 2\%r, 90\%r, 110\%r)$
	p_∞	101325 $[N/m^2]$	—
	T_∞	288.5 $[K]$	—
Geometrical	R_s	0.00839	$\mathcal{TN}(r, 2\%r, 90\%r, 110\%r)$
	R_p	0.00853	$\mathcal{TN}(r, 2\%r, 90\%r, 110\%r)$
	x_s	0.431	$\mathcal{TN}(r, 2\%r, 90\%r, 110\%r)$
	x_p	0.346	$\mathcal{TN}(r, 2\%r, 90\%r, 110\%r)$
	y_s	0.063	$\mathcal{TN}(r, 2\%r, 90\%r, 110\%r)$
	y_p	-0.058	$\mathcal{TN}(r, 2\%r, 90\%r, 110\%r)$
	C_s	-0.432	-
	C_p	0.699	-
	θ_s	-11.607	-
	θ_p	-2.227	-

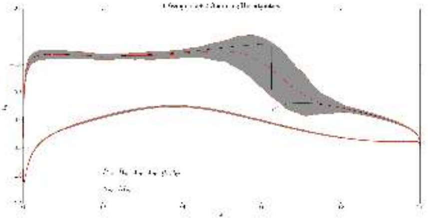


Computation of C_L and pressure coeff. for RAE2822 airfoil



MLMC 5-levels grid hierarchy for the RAE2822 problem.

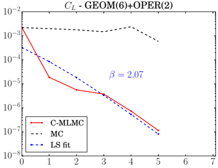
Level	Airfoil nodes	Cells	$\tau(Q_{M_l})[s]$ $(n.cpu)$
$L0$	67	5197	14.4 (18)
$L1$	131	9968	21.4 (22)
$L2$	259	20850	28.8 (28)
$L3$	515	47476	64.0 (36)
$L4$	1027	114857	122.1 (44)
$L5$	2051	283925	314.2 (56)

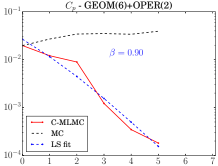


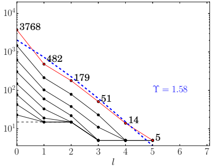
Inviscid model (Euler); SU^2 solver (Stanford) [Pisaroni-N.-Leyland CMAME 2017]

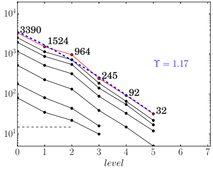
EPFL

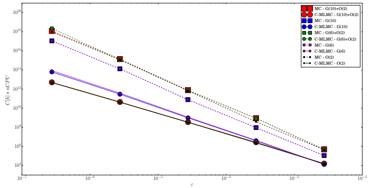
MLMC hierarchies and comparison with MC

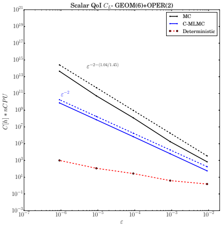


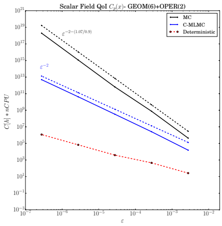




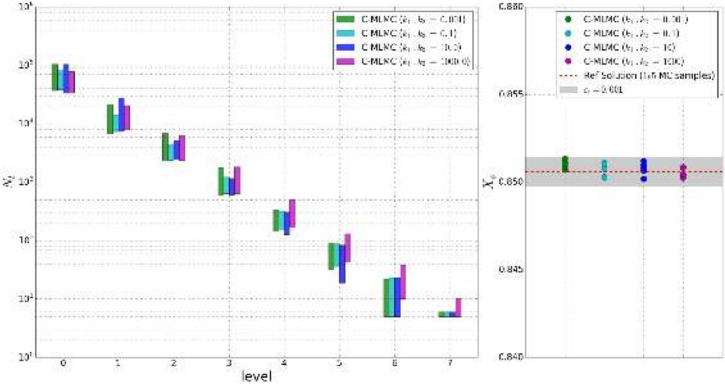








Robustness of C-MLMC estimator



Variability over 10 repetitions of the C-MLMC algorithm for different parameters in the Normal-Gamma prior.

Outline

Beyond expectations: computation of central moments

Goal: compute $\mu_p(Q) = \mathbb{E}[(Q - \mathbb{E}[Q])^p]$

How to apply and tune MLMC in this case?

Let $\vec{Q}_M = \{Q^{(1)}, \ldots, Q^{(M)}\}$ be an iid sample from Q and $\hat{\mu}_p(\vec{Q}_M)$ and estimator for $\mu_p(Q)$. E.g. for $p = 2$ consider the sample variance estimator

$$\hat{\mu}_2(\vec{Q}_M) = \frac{1}{M-1} \sum_{i=1}^{M} \left(Q^{(i)} - \sum_{j=1}^{M} \frac{Q^{(j)}}{M} \right)^2$$

Idea: telescope on $\hat{\mu}_p$

$$\hat{\mu}_p^{MLMC} = \hat{\mu}_p(\vec{Q}_{0,M_0}) + \sum_{\ell=1}^{L} \left(\hat{\mu}_p(\vec{Q}_{\ell,M_\ell}) - \hat{\mu}_p(\vec{Q}_{\ell-1,M_{\ell-1}}) \right)$$

with $(\vec{Q}_{\ell,M_\ell}, \vec{Q}_{\ell-1,M_\ell})$ generated with the same noise, and otherwise independent between levels.

Beyond expectations: computation of central moments

Goal: compute $\mu_p(Q) = \mathbb{E}[(Q - \mathbb{E}[Q])^p]$

How to apply and tune MLMC in this case?

Let $\vec{Q}_M = \{Q^{(1)}, \ldots, Q^{(M)}\}$ be an iid sample from Q and $\hat{\mu}_p(\vec{Q}_M)$ and estimator for $\mu_p(Q)$. E.g. for $p = 2$ consider the sample variance estimator

$$\hat{\mu}_2(\vec{Q}_M) = \frac{1}{M-1} \sum_{i=1}^{M} \left(Q^{(i)} - \sum_{j=1}^{M} \frac{Q^{(j)}}{M} \right)^2$$

Idea: telescope on $\hat{\mu}_p$

$$\hat{\mu}_p^{MLMC} = \hat{\mu}_p(\vec{Q}_{0,M_0}) + \sum_{\ell=1}^{L} \left(\hat{\mu}_p(\vec{Q}_{\ell,M_\ell}) - \hat{\mu}_p(\vec{Q}_{\ell-1,M_{\ell-1}}) \right)$$

with $(\vec{Q}_{\ell,M_\ell}, \vec{Q}_{\ell-1,M_\ell})$ generated with the same noise, and otherwise independent between levels.

Beyond expectations: computation of central moments

Goal: compute $\mu_p(Q) = \mathbb{E}[(Q - \mathbb{E}[Q])^p]$

How to apply and tune MLMC in this case?

Let $\vec{Q}_M = \{Q^{(1)}, \dots, Q^{(M)}\}$ be an iid sample from Q and $\hat{\mu}_p(\vec{Q}_M)$ and estimator for $\mu_p(Q)$. E.g. for $p = 2$ consider the sample variance estimator

$$\hat{\mu}_2(\vec{Q}_M) = \frac{1}{M-1} \sum_{i=1}^{M} \left(Q^{(i)} - \sum_{j=1}^{M} \frac{Q^{(j)}}{M} \right)^2$$

Idea: telescope on $\hat{\mu}_p$

$$\hat{\mu}_p^{MLMC} = \hat{\mu}_p(\vec{Q}_{0,M_0}) + \sum_{\ell=1}^{L} \left(\hat{\mu}_p(\vec{Q}_{\ell,M_\ell}) - \hat{\mu}_p(\vec{Q}_{\ell-1,M_{\ell-1}}) \right)$$

with $(\vec{Q}_{\ell,M_\ell}, \vec{Q}_{\ell-1,M_\ell})$ generated with the same noise, and otherwise independent between levels.

Beyond expectations: computation of central moments

Two main issues:

- It is important that $\hat{\mu}_p(\vec{Q}_M)$ is unbiased to preserve the telescopic property in expectation. (M_ℓ is small on finest levels and the corresponding bias could be large)
- We should be able to estimate $\mathbb{V}\mathrm{ar}[\hat{\mu}_p(\vec{Q}_{\ell,M_\ell}) - \hat{\mu}_p(\vec{Q}_{\ell-1,M_{\ell-1}})]$ to tune the MLMC algorithm

Idea: use *h*-statistics [Pisaroni-Krumscheid-N. 2017]

$$h_p(\vec{Q}_M) : \text{unbiased estimator of } \mu_p(Q) \text{ with minimal variance}$$

(see [Bierig-Chernov 2015-2016] for an alternative approach with biased estimators)

Multilevel estimator: $$h_p^{MLMC} = \sum_{\ell=0}^{L}(h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})), \quad \vec{Q}_{-1,M_0} = \vec{0}$$

Beyond expectations: computation of central moments

Two main issues:

- It is important that $\hat{\mu}_p(\vec{Q}_M)$ is unbiased to preserve the telescopic property in expectation. (M_ℓ is small on finest levels and the corresponding bias could be large)
- We should be able to estimate $\mathbb{V}\mathrm{ar}[\hat{\mu}_p(\vec{Q}_{\ell,M_\ell}) - \hat{\mu}_p(\vec{Q}_{\ell-1,M_{\ell-1}})]$ to tune the MLMC algorithm

Idea: use *h*-statistics [Pisaroni-Krumscheid-N. 2017]

$$h_p(\vec{Q}_M) : \text{unbiased estimator of } \mu_p(Q) \text{ with minimal variance}$$

(see [Bierig-Chernov 2015-2016] for an alternative approach with biased estimators)

Multilevel estimator: $$h_p^{MLMC} = \sum_{\ell=0}^{L}(h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})), \quad \vec{Q}_{-1,M_0} = \vec{0}$$

Beyond expectations: computation of central moments

Two main issues:

- It is important that $\hat{\mu}_p(\vec{Q}_M)$ is unbiased to preserve the telescopic property in expectation. (M_ℓ is small on finest levels and the corresponding bias could be large)
- We should be able to estimate $\mathbb{V}\mathrm{ar}[\hat{\mu}_p(\vec{Q}_{\ell,M_\ell}) - \hat{\mu}_p(\vec{Q}_{\ell-1,M_{\ell-1}})]$ to tune the MLMC algorithm

Idea: use *h*-statistics [Pisaroni-Krumscheid-N. 2017]

$$h_p(\vec{Q}_M) : \text{unbiased estimator of } \mu_p(Q) \text{ with minimal variance}$$

(see [Bierig-Chernov 2015-2016] for an alternative approach with biased estimators)

Multilevel estimator: $$h_p^{MLMC} = \sum_{\ell=0}^{L} (h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})), \quad \vec{Q}_{-1,M_0} = \vec{0}$$

Beyond expectations: computation of central moments

Two main issues:

- It is important that $\hat{\mu}_p(\vec{Q}_M)$ is unbiased to preserve the telescopic property in expectation. (M_ℓ is small on finest levels and the corresponding bias could be large)
- We should be able to estimate $\mathbb{V}\mathrm{ar}[\hat{\mu}_p(\vec{Q}_{\ell,M_\ell}) - \hat{\mu}_p(\vec{Q}_{\ell-1,M_{\ell-1}})]$ to tune the MLMC algorithm

Idea: use h-statistics [Pisaroni-Krumscheid-N. 2017]

$$h_p(\vec{Q}_M) \text{ : unbiased estimator of } \mu_p(Q) \text{ with minimal variance}$$

(see [Bierig-Chernov 2015-2016] for an alternative approach with biased estimators)

Multilevel estimator: $$h_p^{MLMC} = \sum_{\ell=0}^{L} (h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})), \quad \vec{Q}_{-1,M_0} = \vec{0}$$

Computation of central moments

Observe that

$$\mathbb{E}[h_p^{MLMC}] = \sum_{\ell=0}^{L}(\mathbb{E}[h_p(\vec{Q}_{\ell,M_\ell})] - \mathbb{E}[h_p(\vec{Q}_{\ell-1,M_\ell}))]$$

$$= \sum_{\ell=0}^{L}(\mu_p(Q_\ell) - \mu_p(Q_{\ell-1})) = \mu_p(Q_L)$$

$$\mathbb{V}\mathrm{ar}[h_p^{MLMC}] = \sum_{\ell=0}^{L} \underbrace{\mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]}_{=O(M_\ell^{-1})}$$

Definind $V_{\ell,p} = M_\ell \mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]$ we have

Mean squared error: $$\mathrm{MSE}(h_p^{MLMC}) = \underbrace{(\mu_p(Q) - \mu_p(Q_L))^2}_{\mathrm{Bias}^2} + \underbrace{\sum_{\ell=0}^{L} \frac{V_{\ell,p}}{M_\ell}}_{\mathrm{Variance}}$$

Same structure of MSE as for expectation.

Computation of central moments

Observe that

$$\mathbb{E}[h_p^{MLMC}] = \sum_{\ell=0}^{L}(\mathbb{E}[h_p(\vec{Q}_{\ell,M_\ell})] - \mathbb{E}[h_p(\vec{Q}_{\ell-1,M_\ell}))]$$

$$= \sum_{\ell=0}^{L}(\mu_p(Q_\ell) - \mu_p(Q_{\ell-1})) = \mu_p(Q_L)$$

$$\mathbb{V}\mathrm{ar}[h_p^{MLMC}] = \sum_{\ell=0}^{L}\underbrace{\mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]}_{=O(M_\ell^{-1})}$$

Definind $V_{\ell,p} = M_\ell \mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]$ we have

Mean squared error: $$\mathrm{MSE}(h_p^{MLMC}) = \underbrace{(\mu_p(Q) - \mu_p(Q_L))^2}_{\text{Bias}^2} + \underbrace{\sum_{\ell=0}^{L}\frac{V_{\ell,p}}{M_\ell}}_{\text{Variance}}$$

Same structure of MSE as for expectation.

Computation of central moments

Observe that

$$\mathbb{E}[h_p^{MLMC}] = \sum_{\ell=0}^{L} (\mathbb{E}[h_p(\vec{Q}_{\ell,M_\ell})] - \mathbb{E}[h_p(\vec{Q}_{\ell-1,M_\ell})])$$
$$= \sum_{\ell=0}^{L} (\mu_p(Q_\ell) - \mu_p(Q_{\ell-1})) = \mu_p(Q_L)$$

$$\mathbb{V}\mathrm{ar}[h_p^{MLMC}] = \sum_{\ell=0}^{L} \underbrace{\mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]}_{=O(M_\ell^{-1})}$$

Definind $V_{\ell,p} = M_\ell \mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]$ we have

Mean squared error: $$\mathrm{MSE}(h_p^{MLMC}) = \underbrace{(\mu_p(Q) - \mu_p(Q_L))^2}_{\mathrm{Bias}^2} + \underbrace{\sum_{\ell=0}^{L} \frac{V_{\ell,p}}{M_\ell}}_{\mathrm{Variance}}$$

Same structure of MSE as for expectation.

Computation of central moments

Observe that

$$\mathbb{E}[h_p^{MLMC}] = \sum_{\ell=0}^{L} (\mathbb{E}[h_p(\vec{Q}_{\ell,M_\ell})] - \mathbb{E}[h_p(\vec{Q}_{\ell-1,M_\ell}))]$$

$$= \sum_{\ell=0}^{L} (\mu_p(Q_\ell) - \mu_p(Q_{\ell-1})) = \mu_p(Q_L)$$

$$\mathbb{V}\mathrm{ar}[h_p^{MLMC}] = \sum_{\ell=0}^{L} \underbrace{\mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]}_{=O(M_\ell^{-1})}$$

Definind $V_{\ell,p} = M_\ell \mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]$ we have

Mean squared error: $$\mathrm{MSE}(h_p^{MLMC}) = \underbrace{(\mu_p(Q) - \mu_p(Q_L))^2}_{\text{Bias}^2} + \underbrace{\sum_{\ell=0}^{L} \frac{V_{\ell,p}}{M_\ell}}_{\text{Variance}}$$

Same structure of MSE as for expectation.

Computation of central moments

Observe that

$$\mathbb{E}[h_p^{MLMC}] = \sum_{\ell=0}^{L} (\mathbb{E}[h_p(\vec{Q}_{\ell,M_\ell})] - \mathbb{E}[h_p(\vec{Q}_{\ell-1,M_\ell}))]$$

$$= \sum_{\ell=0}^{L} (\mu_p(Q_\ell) - \mu_p(Q_{\ell-1})) = \mu_p(Q_L)$$

$$\mathbb{V}\mathrm{ar}[h_p^{MLMC}] = \sum_{\ell=0}^{L} \underbrace{\mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]}_{=O(M_\ell^{-1})}$$

Definind $V_{\ell,p} = M_\ell \mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]$ we have

Mean squared error: $$\mathrm{MSE}(h_p^{MLMC}) = \underbrace{(\mu_p(Q) - \mu_p(Q_L))^2}_{\text{Bias}^2} + \underbrace{\sum_{\ell=0}^{L} \frac{V_{\ell,p}}{M_\ell}}_{\text{Variance}}$$

Same structure of MSE as for expectation.

Computation of central moments

Observe that

$$\mathbb{E}[h_p^{MLMC}] = \sum_{\ell=0}^{L} (\mathbb{E}[h_p(\vec{Q}_{\ell,M_\ell})] - \mathbb{E}[h_p(\vec{Q}_{\ell-1,M_\ell}))]$$

$$= \sum_{\ell=0}^{L} (\mu_p(Q_\ell) - \mu_p(Q_{\ell-1})) = \mu_p(Q_L)$$

$$\mathbb{V}\mathrm{ar}[h_p^{MLMC}] = \sum_{\ell=0}^{L} \underbrace{\mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]}_{=O(M_\ell^{-1})}$$

Definind $V_{\ell,p} = M_\ell \mathbb{V}\mathrm{ar}[h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})]$ we have

Mean squared error: $$\mathrm{MSE}(h_p^{MLMC}) = \underbrace{(\mu_p(Q) - \mu_p(Q_L))^2}_{\text{Bias}^2} + \underbrace{\sum_{\ell=0}^{L} \frac{V_{\ell,p}}{M_\ell}}_{\text{Variance}}$$

Same structure of MSE as for expectation.

Computation of central moments

Complexity result for $h_\ell = h_0\delta^\ell$, $\delta \in (0,1)$

Assume $\mu_{2p}(Q_\ell) < \infty$ for all ℓ and there exist $\alpha, \beta, \gamma > 0$, $2\alpha \geq \min\{\beta, d\gamma\}$ s.t.

- $|\mu_p(Q) - \mu_p(Q_\ell)| = \mathcal{O}(h_\ell^\alpha)$,
- $V_{\ell,p} = O(h_\ell^\beta)$,
- $C_\ell = \mathrm{Cost}(Q_\ell^{(i,\ell)}, Q_{\ell-1}^{(i,\ell)}) = \mathcal{O}(h_\ell^{-d\gamma})$,

Then, taking $L = \mathcal{O}(tol^{\frac{1}{\alpha}})$ and $M_\ell = \left\lceil tol^{-2}\sqrt{\frac{V_{\ell,p}}{C_\ell}}\left(\sum_{k=0}^{L}\sqrt{C_k V_{k,p}}\right)\right\rceil$ leads to

$$\mathrm{MSE}(h_p^{MLMC}) \lesssim tol^2 \quad \text{and} \quad \mathrm{Cost}(h_p^{MLMC}, \mathrm{tol}) \lesssim \begin{cases} tol^{-2}, & \beta > d\gamma \\ tol^{-2}|\log(tol)|^2, & \beta = d\gamma \\ tol^{-2-\frac{d\gamma-\beta}{\alpha}}, & \beta < d\gamma \end{cases}$$

Computation of central moments

Technical difficulty: how to estimate the variances $V_{\ell,p}$ (needed for optimal allocation and error control)

Define $\vec{X}^{+}_{\ell,M_\ell} = \vec{Q}_{\ell,M_\ell} + \vec{Q}_{\ell-1,M_\ell}$, $\vec{X}^{-}_{\ell,M_\ell} = \vec{Q}_{\ell,M_\ell} - \vec{Q}_{\ell-1,M_\ell}$
$\Delta_\ell h_p = h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})$ can be expressed as a power sum

$$\Delta_\ell h_p = \sum_{a+b\leq p} S_{a,b}(\vec{X}^{+}_{\ell,M_\ell}, \vec{X}^{-}_{\ell,M_\ell}), \qquad S_{a,b}(\vec{X},\vec{Y}) = \sum_i (X^{(i)})^a (Y^{(i)})^b$$

Unbiased estimators $\hat{V}_{\ell,p}$ of $V_{\ell,p}$ can be computed in closed form starting from the power terms $S_{a,b}(\vec{X}^{+}_{\ell,M_\ell}, \vec{X}^{-}_{\ell,M_\ell})$ [Pisaroni-Krumscheid-N. 2017].

Computation of central moments

Technical difficulty: how to estimate the variances $V_{\ell,p}$ (needed for optimal allocation and error control)

Define $\vec{X}^{+}_{\ell,M_\ell} = \vec{Q}_{\ell,M_\ell} + \vec{Q}_{\ell-1,M_\ell}$, $\vec{X}^{-}_{\ell,M_\ell} = \vec{Q}_{\ell,M_\ell} - \vec{Q}_{\ell-1,M_\ell}$

$\Delta_\ell h_p = h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})$ can be expressed as a power sum

$$\Delta_\ell h_p = \sum_{a+b\leq p} S_{a,b}(\vec{X}^{+}_{\ell,M_\ell}, \vec{X}^{-}_{\ell,M_\ell}), \qquad S_{a,b}(\vec{X},\vec{Y}) = \sum_i (X^{(i)})^a (Y^{(i)})^b$$

Unbiased estimators $\hat{V}_{\ell,p}$ of $V_{\ell,p}$ can be computed in closed form starting from the power terms $S_{a,b}(\vec{X}^{+}_{\ell,M_\ell}, \vec{X}^{-}_{\ell,M_\ell})$ [Pisaroni-Krumscheid-N. 2017].

Computation of central moments

Technical difficulty: how to estimate the variances $V_{\ell,p}$ (needed for optimal allocation and error control)

Define $\vec{X}^+_{\ell,M_\ell} = \vec{Q}_{\ell,M_\ell} + \vec{Q}_{\ell-1,M_\ell}$, $\vec{X}^-_{\ell,M_\ell} = \vec{Q}_{\ell,M_\ell} - \vec{Q}_{\ell-1,M_\ell}$
$\Delta_\ell h_p = h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})$ can be expressed as a power sum

$$\Delta_\ell h_p = \sum_{a+b\leq p} S_{a,b}(\vec{X}^+_{\ell,M_\ell}, \vec{X}^-_{\ell,M_\ell}), \qquad S_{a,b}(\vec{X}, \vec{Y}) = \sum_i (X^{(i)})^a (Y^{(i)})^b$$

Unbiased estimators $\hat{V}_{\ell,p}$ of $V_{\ell,p}$ can be computed in closed form starting from the power terms $S_{a,b}(\vec{X}^+_{\ell,M_\ell}, \vec{X}^-_{\ell,M_\ell})$ [Pisaroni-Krumscheid-N. 2017].

Beyond expectations: CDF, quantiles, and more

The cumulative distribution function (CDF) of Q can be seen as a parametric expectation

$$F(\theta) = \mathbb{E}[\phi(\theta, Q)], \qquad \phi(\theta, Q) = \mathbb{1}_{\{Q \leq \theta\}}$$

One could apply MLMC on many values θ_i (using the same sample of Q) and interpolate.

Problem: $\phi(\theta, Q)$ is not smooth! the variance of the differences, $V_\ell = \mathbb{V}\mathrm{ar}[\phi(\theta, Q_\ell) - \phi(\theta, Q_{\ell-1})]$ will decay slowly. No much gain in using MLMC vs MC.

Remedies:

- [Giles-Nagapetyan-Ritter 2015, 2017] smoothing: $F_\epsilon(\theta) = \mathbb{E}[\phi_\epsilon(\theta, Q)]$. Technical difficulty: ϵ should depend on the required tolerance $\rightsquigarrow$ difficult tuning of MLMC
- [Bierig-Chernov 2016] approximate F or pdf based on moments
- [Krumscheid-N. 2017] anti-derivative approach: $F(\theta) = \Phi'(\theta)$ with $\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)]$ and $\phi(\theta, \cdot)$ Lipschitz continuous.

Beyond expectations: CDF, quantiles, and more

The cumulative distribution function (CDF) of Q can be seen as a parametric expectation

$$F(\theta) = \mathbb{E}[\phi(\theta, Q)], \qquad \phi(\theta, Q) = \mathbb{1}_{\{Q \leq \theta\}}$$

One could apply MLMC on many values θ_i (using the same sample of Q) and interpolate.

Problem: $\phi(\theta, Q)$ is not smooth! the variance of the differences, $V_\ell = \mathbb{V}\mathrm{ar}[\phi(\theta, Q_\ell) - \phi(\theta, Q_{\ell-1})]$ will decay slowly. No much gain in using MLMC vs MC.

Remedies:

- [Giles-Nagapetyan-Ritter 2015, 2017] smoothing: $F_\epsilon(\theta) = \mathbb{E}[\phi_\epsilon(\theta, Q)]$. Technical difficulty: ϵ should depend on the required tolerance $\rightsquigarrow$ difficult tuning of MLMC
- [Bierig-Chernov 2016] approximate F or pdf based on moments
- [Krumscheid-N. 2017] anti-derivative approach: $F(\theta) = \Phi'(\theta)$ with $\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)]$ and $\phi(\theta, \cdot)$ Lipschitz continuous.

Beyond expectations: CDF, quantiles, and more

The cumulative distribution function (CDF) of Q can be seen as a parametric expectation

$$F(\theta) = \mathbb{E}[\phi(\theta, Q)], \qquad \phi(\theta, Q) = \mathbb{1}_{\{Q \leq \theta\}}$$

One could apply MLMC on many values θ_i (using the same sample of Q) and interpolate.

Problem: $\phi(\theta, Q)$ is not smooth! the variance of the differences, $V_\ell = \mathbb{V}\mathrm{ar}[\phi(\theta, Q_\ell) - \phi(\theta, Q_{\ell-1})]$ will decay slowly. No much gain in using MLMC vs MC.

Remedies:

- [Giles-Nagapetyan-Ritter 2015, 2017] smoothing: $F_\epsilon(\theta) = \mathbb{E}[\phi_\epsilon(\theta, Q)]$. Technical difficulty: ϵ should depend on the required tolerance $\rightsquigarrow$ difficult tuning of MLMC
- [Bierig-Chernov 2016] approximate F or pdf based on moments
- [Krumscheid-N. 2017] anti-derivative approach: $F(\theta) = \Phi'(\theta)$ with $\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)]$ and $\phi(\theta, \cdot)$ Lipschitz continuous.

Anti-derivative approach to CDF computation

For a given $\tau \in (0,1)$ define

$$\Phi_\tau(\theta) = \mathbb{E}[\phi_\tau(\theta, Q)], \qquad \phi_\tau(\theta, Q) = \theta + \frac{1}{1+\tau}(Q-\theta)_+$$

Then (assuming $F \in C^1$)

$$F(\theta) = (1-\tau)\Phi'_\tau(\theta) + \tau$$

and MLMC can be effectively used to approximate $\Phi_\tau(\theta)$ and its derivatives.

Moreover, from the approximation of Φ_τ and its derivatives we can get for free

- pdf: $p(\theta) = F'(\theta) = (1-\tau)\Phi''_\tau(\theta)$
- τ-quantile: $q_\tau = \inf\{\theta : F(\theta) \geq \tau\} = \text{argmin}_{\theta\in\mathbb{R}} \Phi_\tau(\theta)$
- Conditional Value at Risk

$$CVaR_\tau = \frac{1}{1-\tau}\int_{q_\tau}^{\infty} x dF(x) = \min_{\theta\in\mathbb{R}} \Phi_\tau(\theta)$$

Anti-derivative approach to CDF computation

For a given $\tau \in (0,1)$ define

$$\Phi_\tau(\theta) = \mathbb{E}[\phi_\tau(\theta, Q)], \qquad \phi_\tau(\theta, Q) = \theta + \frac{1}{1+\tau}(Q-\theta)_+$$

Then (assuming $F \in C^1$)

$$F(\theta) = (1-\tau)\Phi'_\tau(\theta) + \tau$$

and MLMC can be effectively used to approximate $\Phi_\tau(\theta)$ and its derivatives.

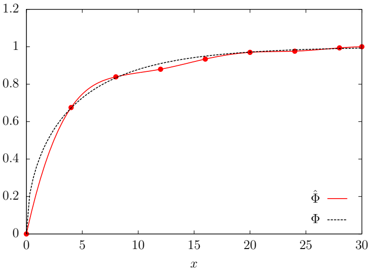
Moreover, from the approximation of Φ_τ and its derivatives we can get for free

- pdf: $p(\theta) = F'(\theta) = (1-\tau)\Phi''_\tau(\theta)$
- τ-quantile: $q_\tau = \inf\{\theta : F(\theta) \geq \tau\} = \mathrm{argmin}_{\theta\in\mathbb{R}} \Phi_\tau(\theta)$
- Conditional Value at Risk

$$CVaR_\tau = \frac{1}{1-\tau}\int_{q_\tau}^{\infty} x dF(x) = \min_{\theta\in\mathbb{R}} \Phi_\tau(\theta)$$

Computing parametric expectations by MLMC

Goal: given $\phi(\theta, Q)$, approximate $\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)]$ and its derivatives uniformly in Θ. **Notation**: $\Phi_\ell(\theta) := \mathbb{E}[\phi(\theta, Q_\ell)]$.



Interpolation approach:

- introduce a grid $\vec{\theta} = \{\theta_1, \dots, \theta_n\} \subset \Theta$
- compute $\Phi^{MLMC}(\theta_j)$, $j = 1, \dots, n$ by MLMC (same sample of Q_ℓ for every θ_j)
- Interpolate values $\hat{\Phi}^{MLMC}(\vec{\theta}) = \{\hat{\Phi}^{MLMC}(\theta_j)\}_{j=1}^n$

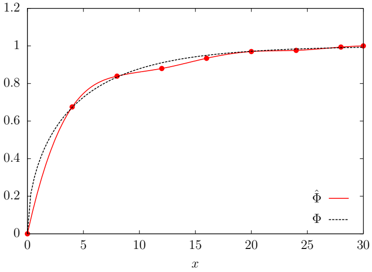
$$\hat{\Phi}^{MLMC} = \mathcal{I}_n(\hat{\Phi}^{MLMC}(\vec{\theta}))$$

e.g. by spline or polynomial interpolation

Eventually, compute also derivatives $\dfrac{d^m \hat{\Phi}^{MLMC}}{d\theta^m}$

Computing parametric expectations by MLMC

Goal: given $\phi(\theta, Q)$, approximate $\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)]$ and its derivatives uniformly in Θ. **Notation**: $\Phi_\ell(\theta) := \mathbb{E}[\phi(\theta, Q_\ell)]$.



Interpolation approach:

- introduce a grid $\vec{\theta} = \{\theta_1, \dots, \theta_n\} \subset \Theta$
- compute $\Phi^{MLMC}(\theta_j)$, $j = 1, \dots, n$ by MLMC (same sample of Q_ℓ for every θ_j)
- Interpolate values $\hat{\Phi}^{MLMC}(\vec{\theta}) = \{\hat{\Phi}^{MLMC}(\theta_j)\}_{j=1}^n$

$$\hat{\Phi}^{MLMC} = \mathcal{I}_n(\hat{\Phi}^{MLMC}(\vec{\theta}))$$

e.g. by spline or polynomial interpolation

Eventually, compute also derivatives $\dfrac{d^m \hat{\Phi}^{MLMC}}{d\theta^m}$

Error splitting

Define the mean squared error: $\mathrm{MSE}(\hat{\Phi}^{MLMC}) = \mathbb{E}[\sup_{\theta\in\Theta} |\Phi(\theta) - \hat{\Phi}^{MLMC}(\theta)|^2]$

Error splitting

$$\mathrm{MSE}(\hat{\Phi}^{MLMC}) \lesssim \underbrace{\|\Phi - \mathcal{I}_n\Phi(\vec{\theta})\|^2_\infty}_{\text{interp. error}} + \underbrace{\|\Phi(\vec{\theta}) - \Phi_L(\vec{\theta})\|^2_\infty}_{\text{discret. error}} + \log(n) \underbrace{\sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}}_{\text{statistical error}}$$

with $V_\ell = \mathbb{E}[\|\Delta\phi(\vec{\theta}, Q_\ell) - \mathbb{E}[\Delta\phi(\vec{\theta}, Q_\ell)]\|^2_{\ell^\infty}]$

All terms (and constants) can be estimated in practice... but rather painful. Optimization of MLMC based on estimators $\hat{V}_\ell$ of V_ℓ [AyoulGuilmard-Ganesh-Krumscheid-N.-Pisaroni in preparation]

Complexity analysis for the error on Φ and its derivatives available in [Krumscheid-N. 2017].

Error splitting

Define the mean squared error: $\mathrm{MSE}(\hat{\Phi}^{MLMC}) = \mathbb{E}[\sup_{\theta\in\Theta} |\Phi(\theta) - \hat{\Phi}^{MLMC}(\theta)|^2]$

Error splitting

$$\mathrm{MSE}(\hat{\Phi}^{MLMC}) \lesssim \underbrace{\|\Phi - \mathcal{I}_n\Phi(\vec{\theta})\|^2_\infty}_{\text{interp. error}} + \underbrace{\|\Phi(\vec{\theta}) - \Phi_L(\vec{\theta})\|^2_\infty}_{\text{discret. error}} + \log(n) \underbrace{\sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}}_{\text{statistical error}}$$

with $V_\ell = \mathbb{E}[\|\Delta\phi(\vec{\theta}, Q_\ell) - \mathbb{E}[\Delta\phi(\vec{\theta}, Q_\ell)]\|^2_{\ell^\infty}]$

All terms (and constants) can be estimated in practice... but rather painful.
Optimization of MLMC based on estimators $\hat{V}_\ell$ of V_ℓ
[AyoulGuilmard-Ganesh-Krumscheid-N.-Pisaroni in preparation]

Complexity analysis for the error on Φ and its derivatives available in [Krumscheid-N. 2017].

Error splitting

Define the mean squared error: $\mathrm{MSE}(\hat{\Phi}^{MLMC}) = \mathbb{E}[\sup_{\theta\in\Theta} |\Phi(\theta) - \hat{\Phi}^{MLMC}(\theta)|^2]$

Error splitting

$$\mathrm{MSE}(\hat{\Phi}^{MLMC}) \lesssim \underbrace{\|\Phi - \mathcal{I}_n\Phi(\vec{\theta})\|^2_\infty}_{\text{interp. error}} + \underbrace{\|\Phi(\vec{\theta}) - \Phi_L(\vec{\theta})\|^2_\infty}_{\text{discret. error}} + \log(n) \underbrace{\sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}}_{\text{statistical error}}$$

with $V_\ell = \mathbb{E}[\|\Delta\phi(\vec{\theta}, Q_\ell) - \mathbb{E}[\Delta\phi(\vec{\theta}, Q_\ell)]\|^2_{\ell^\infty}]$

All terms (and constants) can be estimated in practice... but rather painful. Optimization of MLMC based on estimators $\hat{V}_\ell$ of V_ℓ
[AyoulGuilmard-Ganesh-Krumscheid-N.-Pisaroni in preparation]

Complexity analysis for the error on Φ and its derivatives available in [Krumscheid-N. 2017].

EPFL

Error splitting

Define the mean squared error: $\mathrm{MSE}(\hat{\Phi}^{MLMC}) = \mathbb{E}[\sup_{\theta\in\Theta} |\Phi(\theta) - \hat{\Phi}^{MLMC}(\theta)|^2]$

Error splitting

$$\mathrm{MSE}(\hat{\Phi}^{MLMC}) \lesssim \underbrace{\|\Phi - \mathcal{I}_n\Phi(\vec{\theta})\|^2_\infty}_{\text{interp. error}} + \underbrace{\|\Phi(\vec{\theta}) - \Phi_L(\vec{\theta})\|^2_\infty}_{\text{discret. error}} + \log(n) \underbrace{\sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}}_{\text{statistical error}}$$

with $V_\ell = \mathbb{E}[\|\Delta\phi(\vec{\theta}, Q_\ell) - \mathbb{E}[\Delta\phi(\vec{\theta}, Q_\ell)]\|^2_{\ell^\infty}]$

All terms (and constants) can be estimated in practice... but rather painful.
Optimization of MLMC based on estimators $\hat{V}_\ell$ of V_ℓ
[AyoulGuilmard-Ganesh-Krumscheid-N.-Pisaroni in preparation]

Complexity analysis for the error on Φ and its derivatives available in [Krumscheid-N. 2017].

Example – risk averse optimization

$$\min_{x \in X} \mathcal{R}(Q(x)), \qquad X\text{: feasible design space}$$

$\mathcal{R}$: risk measure

Examples

- $\mathcal{R}(Q) = \mathbb{E}[Q]$ (risk neutral)
- $\mathcal{R}(Q) = \mathbb{E}[Q] \pm \alpha\, \mathrm{std}[Q]$
- $\mathcal{R}(Q) = q_\tau[Q]$ (τ-quantile)
- $\mathcal{R}(Q) = CVaR_\tau[Q]$

Example – risk averse optimization

$$\min_{x \in X} \mathcal{R}(Q(x)), \qquad X\text{: feasible design space}$$

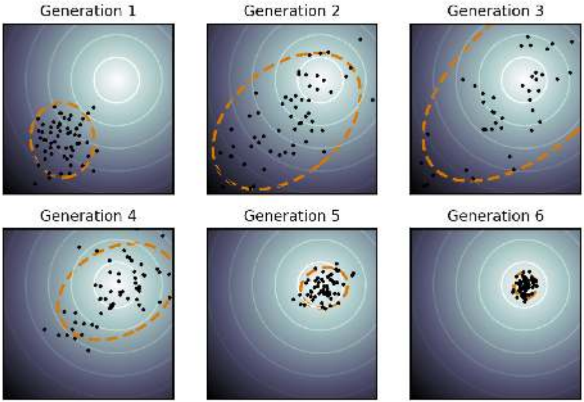
$\mathcal{R}$: risk measure

Examples

- $\mathcal{R}(Q) = \mathbb{E}[Q]$ (risk neutral)
- $\mathcal{R}(Q) = \mathbb{E}[Q] \pm \alpha \, \mathrm{std}[Q]$
- $\mathcal{R}(Q) = q_\tau[Q]$ (τ-quantile)
- $\mathcal{R}(Q) = CVaR_\tau[Q]$

Combining MLMC with CMA-ES

Optimization done by Covariance Matrix Adaptation Evolutionary Algorithm (CMA-ES)

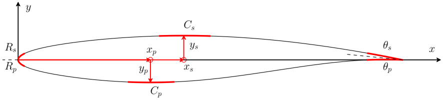


For each individual at each generation, risk measure computed by MLMC.

Airfoil optimization under operating uncertainties

$$\begin{cases} \min_{x \in X} & \mathcal{R}\left[C_D(x)\right] \\ s.t & C_L(x) = C_L^*, \quad \text{thickness constraint} \end{cases}$$

x: airfoil shape – PARSEC parameters $(R_s, R_p, x_s, x_p, C_s, C_p, \theta_s, \theta_p)$

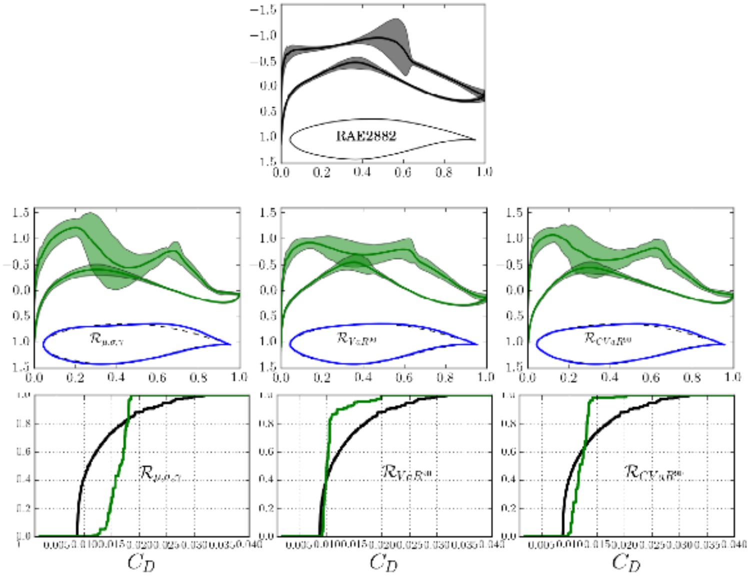


$\mathcal{R}_{\mu,\sigma}\left[C_D(x)\right]$	$\mu_{C_D}(x) + \sigma_{C_D}(x)$
$\mathcal{R}_{\mu,\sigma,\gamma}\left[C_D(x)\right]$	$\mu_{C_D}(x) + \sigma_{C_D}(x) + \mu_{C_D}(x) \cdot \gamma_{C_D}(x)$
$\mathcal{R}_{VaR^{90}}\left[C_D(x)\right]$	$VaR^{90}_{C_D}(x)$
$\mathcal{R}_{CVaR^{90}}\left[C_D(x)\right]$	$CVaR^{90}_{C_D}(x)$

	Quantity	**Reference** (r)	**Uncertainty**
Operating parameters	C_L	0.5	–
	M_∞	0.75	$\mathcal{B}(2, 2, 0.1, M_\infty - 0.05)$
	Re_c	$6.5 \cdot 10^6$	–
	p_∞ $[Pa]$	101325	–
	T_∞ $[K]$	288.5	–

Model: steady state Euler + boundary layer equation (MSES software)

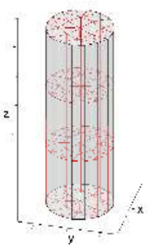
Deterministic versus Robust optimization



Outline

Multi Index Monte Carlo method

Often, the computational model involves several discretization parameters (e.g. spatial mesh size, time step, domain truncation, model simplification, etc.)



numerical solution: $u_{\vec{h}}, \quad \vec{h} = (h^{(1)}, \dots, h^{(d)})$

- Introduce sequences of refined discretizations: $h_0^{(i)} > h_1^{(i)} > \dots > h_{L_i}^{(i)}$
- For $\vec{\ell} = (\ell_1, \dots, \ell_d)$, denote $Q_{\vec{\ell}} = Q(u_{h^{(1)}_{\ell_1}, \dots, h^{(d)}_{\ell_d}})$
- Difference operators

$$\Delta_j Q_{\vec{\ell}} = \begin{cases} Q_{\vec{\ell}} - Q_{\vec{\ell}-\vec{e}_j}, & \text{if } \ell_j > 0 \\ Q_{\vec{\ell}}, & \text{if } \ell_j = 0 \end{cases}$$

$$\Delta Q_{\vec{\ell}} = \bigotimes_{j=1}^{d} \Delta_j Q_{\vec{\ell}} = \sum_{\vec{j} \in \{0,1\}^d} (-1)^{|\vec{j}|} Q_{\vec{\ell}-\vec{j}}$$

Multi Index Monte Carlo method

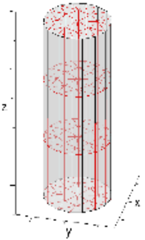
Often, the computational model involves several discretization parameters (e.g. spatial mesh size, time step, domain truncation, model simplification, etc.)

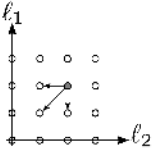
numerical solution: $u_{\vec{h}}$, $\quad \vec{h} = (h^{(1)}, \dots, h^{(d)})$

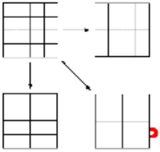
- Introduce sequences of refined discretizations: $h_0^{(i)} > h_1^{(i)} > \dots > h_{L_i}^{(i)}$
- For $\vec{\ell} = (\ell_1, \dots, \ell_d)$, denote $Q_{\vec{\ell}} = Q(u_{h^{(1)}_{\ell_1}, \dots, h^{(d)}_{\ell_d}})$
- Difference operators

$$\Delta_j Q_{\vec{\ell}} = \begin{cases} Q_{\vec{\ell}} - Q_{\vec{\ell} - \vec{e}_j}, & \text{if } \ell_j > 0 \\ Q_{\vec{\ell}}, & \text{if } \ell_j = 0 \end{cases}$$

$$\Delta Q_{\vec{\ell}} = \bigotimes_{j=1}^{d} \Delta_j Q_{\vec{\ell}} = \sum_{\vec{j} \in \{0,1\}^d} (-1)^{|\vec{j}|} Q_{\vec{\ell} - \vec{j}}$$







Multi Index Monte Carlo method

Telescopic formula: given finest discretization level $\vec{L} = (L_1, \ldots, L_d)$

$$\mathbb{E}[Q_{\vec{L}}] = \sum_{\vec{\ell} \leq \vec{L}} \mathbb{E}[\Delta Q_{\vec{\ell}}]$$

Multi Index idea: compute each expectation independently

$$\hat{\mu}_{\vec{L}}^{MIMC} = \sum_{\vec{\ell} \leq \vec{L}} \frac{1}{M_{\vec{\ell}}} \sum_{i=1}^{M_{\vec{\ell}}} \Delta Q_{\vec{\ell}}^{(i,\vec{\ell})}$$

Further sparsification: often the set $\{\vec{\ell} \leq \vec{L}\}$ is not the optimal one. Optimized index sets $\mathcal{I} \subset \mathbb{N}^d$ can lead to substantial improvement

$$\mu_{\mathcal{I}}^{MIMC} = \sum_{\vec{\ell} \in \mathcal{I}} \frac{1}{M_{\vec{\ell}}} \sum_{i=1}^{M_{\vec{\ell}}} \Delta Q_{\vec{\ell}}^{(i,\vec{\ell})}$$

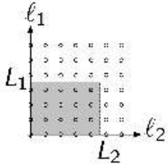
Multi Index Monte Carlo method

Telescopic formula: given finest discretization level $\vec{L} = (L_1, \dots, L_d)$

$$\mathbb{E}[Q_{\vec{L}}] = \sum_{\vec{\ell} \leq \vec{L}} \mathbb{E}[\Delta Q_{\vec{\ell}}]$$

Multi Index idea: compute each expectation independently

$$\hat{\mu}_{\vec{L}}^{MIMC} = \sum_{\vec{\ell} \leq \vec{L}} \frac{1}{M_{\vec{\ell}}} \sum_{i=1}^{M_{\vec{\ell}}} \Delta Q_{\vec{\ell}}^{(i,\vec{\ell})}$$



Further sparsification: often the set $\{\vec{\ell} \leq \vec{L}\}$ is not the optimal one. Optimized index sets $\mathcal{I} \subset \mathbb{N}^d$ can lead to substantial improvement

$$\mu_{\mathcal{I}}^{MIMC} = \sum_{\vec{\ell} \in \mathcal{I}} \frac{1}{M_{\vec{\ell}}} \sum_{i=1}^{M_{\vec{\ell}}} \Delta Q_{\vec{\ell}}^{(i,\vec{\ell})}$$

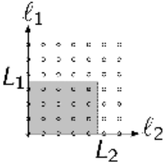
Multi Index Monte Carlo method

Telescopic formula: given finest discretization level $\vec{L} = (L_1, \dots, L_d)$

$$\mathbb{E}[Q_{\vec{L}}] = \sum_{\vec{\ell} \leq \vec{L}} \mathbb{E}[\Delta Q_{\vec{\ell}}]$$

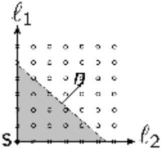
Multi Index idea: compute each expectation independently

$$\hat{\mu}_{\vec{L}}^{MIMC} = \sum_{\vec{\ell} \leq \vec{L}} \frac{1}{M_{\vec{\ell}}} \sum_{i=1}^{M_{\vec{\ell}}} \Delta Q_{\vec{\ell}}^{(i,\vec{\ell})}$$



Further sparsification: often the set $\{\vec{\ell} \leq \vec{L}\}$ is not the optimal one. Optimized index sets $\mathcal{I} \subset \mathbb{N}^d$ can lead to substantial improvement

$$\mu_{\mathcal{I}}^{MIMC} = \sum_{\vec{\ell} \in \mathcal{I}} \frac{1}{M_{\vec{\ell}}} \sum_{i=1}^{M_{\vec{\ell}}} \Delta Q_{\vec{\ell}}^{(i,\vec{\ell})}$$



Complexity analysis

Assume $h_{\ell_i}^{(i)} = h_0^{(i)}\sigma_i^{\ell_i}$, $\sigma_i > 1$ and

- $|\mathbb{E}[\Delta Q_{\vec{\ell}}]| \lesssim \prod_{j=1}^{d} h_{\ell_j}^{\alpha_i}$
- $\mathbb{V}\mathrm{ar}[\Delta Q_{\vec{\ell}}] \lesssim \prod_{j=1}^{d} h_{\ell_j}^{\beta_i}$
- $Cost(\Delta Q_{\vec{\ell}}) \lesssim \prod_i h_{\ell_j}^{-\gamma_i}$

These assumptions require some type of "mixed regularity".

Then, setting $n_i = \log(\sigma_i)(\alpha_i + \frac{\gamma_i - \beta_i}{2})$, the optimal sets are

$$\mathcal{I}_L = \{\vec{\ell} \in \mathbb{N}^d : \ \vec{\ell}\cdot\vec{n} \leq L\}$$

Complexity analysis [HajiAli-N.-Tempone 2015]

Under the above assumptions, for any $tol > 0$ there exist L and $\{M_{\vec{\ell}}\}_{\vec{\ell}\in\mathcal{I}_L}$ such that $MSE(\mu_{\mathcal{I}_L}^{MIMC}) \leq tol^2$ and

$$W(\mu_{\mathcal{I}_L}^{MIMC}) \lesssim \begin{cases} tol^{-2}, & \text{if } \beta_j > \gamma_j, \ \forall j \\ tol^{-2-\max_j \frac{\gamma_j-\beta_j}{\alpha_j}} |\log tol|^p, & \text{otherwise} \end{cases}$$

with p depending on $\#\{j : \frac{\gamma_j-\beta_j}{\alpha_j} = \max_k \frac{\gamma_k-\beta_k}{\alpha_k}\}$

Complexity analysis

Assume $h^{(i)}_{\ell_i} = h^{(i)}_0 \sigma_i^{\ell_i}$, $\sigma_i > 1$ and

- $|\mathbb{E}[\Delta Q_{\vec{\ell}}]| \lesssim \prod_{j=1}^d h_{\ell_j}^{\alpha_i}$
- $\mathbb{V}\mathrm{ar}[\Delta Q_{\vec{\ell}}] \lesssim \prod_{j=1}^d h_{\ell_j}^{\beta_i}$
- $Cost(\Delta Q_{\vec{\ell}}) \lesssim \prod_i h_{\ell_j}^{-\gamma_i}$

These assumptions require some type of "mixed regularity".

Then, setting $n_i = \log(\sigma_i)(\alpha_i + \frac{\gamma_i - \beta_i}{2})$, the optimal sets are

$$\mathcal{I}_L = \{\vec{\ell} \in \mathbb{N}^d : \ \vec{\ell} \cdot \vec{n} \leq L\}$$

Complexity analysis [HajiAli-N.-Tempone 2015]

Under the above assumptions, for any $tol > 0$ there exist L and $\{M_{\vec{\ell}}\}_{\vec{\ell} \in \mathcal{I}_L}$ such that $MSE(\mu^{MIMC}_{\mathcal{I}_L}) \leq tol^2$ and

$$W(\mu^{MIMC}_{\mathcal{I}_L}) \lesssim \begin{cases} tol^{-2}, & \text{if } \beta_j > \gamma_j, \ \forall j \\ tol^{-2-\max_j \frac{\gamma_j - \beta_j}{\alpha_j}} |\log tol|^p, & \text{otherwise} \end{cases}$$

with p depending on $\#\{j : \frac{\gamma_j - \beta_j}{\alpha_j} = \max_k \frac{\gamma_k - \beta_k}{\alpha_k}\}$

Complexity analysis

Assume $h^{(i)}_{\ell_i} = h^{(i)}_0 \sigma_i^{\ell_i}$, $\sigma_i > 1$ and

- $|\mathbb{E}[\Delta Q_{\vec\ell}]| \lesssim \prod_{j=1}^d h_{\ell_j}^{\alpha_i}$
- $\mathbb{V}\mathrm{ar}[\Delta Q_{\vec\ell}] \lesssim \prod_{j=1}^d h_{\ell_j}^{\beta_i}$
- $Cost(\Delta Q_{\vec\ell}) \lesssim \prod_i h_{\ell_j}^{-\gamma_i}$

These assumptions require some type of "mixed regularity".

Then, setting $n_i = \log(\sigma_i)(\alpha_i + \frac{\gamma_i - \beta_i}{2})$, the optimal sets are

$$\mathcal{I}_L = \{\vec\ell \in \mathbb{N}^d : \ \ \vec\ell \cdot \vec n \leq L\}$$

Complexity analysis [HajiAli-N.-Tempone 2015]

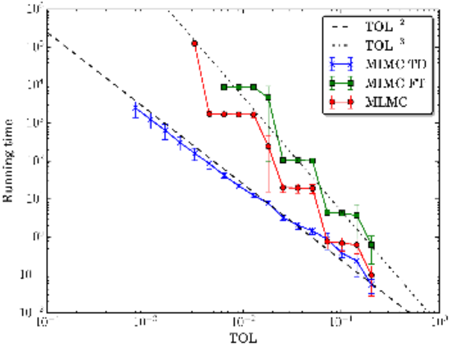
Under the above assumptions, for any $tol > 0$ there exist L and $\{M_{\vec\ell}\}_{\vec\ell \in \mathcal{I}_L}$ such that $MSE(\mu^{MIMC}_{\mathcal{I}_L}) \leq tol^2$ and

$$W(\mu^{MIMC}_{\mathcal{I}_L}) \lesssim \begin{cases} tol^{-2}, & \text{if } \beta_j > \gamma_j, \ \ \forall j \\ tol^{-2-\max_j \frac{\gamma_j - \beta_j}{\alpha_j}} |\log tol|^p, & \text{otherwise} \end{cases}$$

with p depending on $\#\{j : \frac{\gamma_j - \beta_j}{\alpha_j} = \max_k \frac{\gamma_k - \beta_k}{\alpha_k}\}$

Numerical test

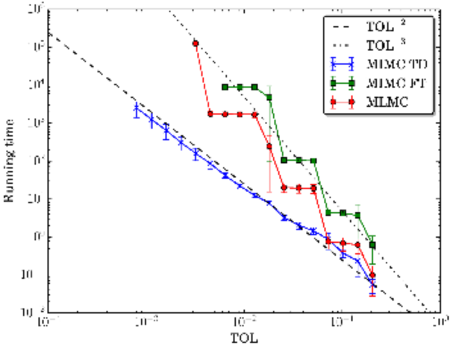
Elliptic equation in 3D with random coefficient and forcing term. Discretization parameters: mesh sizes in the 3 variables (x, y, z) separately.



MIMC has been used also for particle systems (time discretization + n. of particles) [HajiAli-Tempone 2017], nested Monte Carlo simulations [Giles 2015], space-time Zakai type SPDEs [Giles-Reisinger 2016].

Numerical test

Elliptic equation in 3D with random coefficient and forcing term. Discretization parameters: mesh sizes in the 3 variables (x, y, z) separately.



MIMC has been used also for particle systems (time discretization + n. of particles) [HajiAli-Tempone 2017], nested Monte Carlo simulations [Giles 2015], space-time Zakai type SPDEs [Giles-Reisinger 2016].