Multi-level Monte Carlo methods in Uncertainty Quantification

Fabio Nobile

CSQI - Institute of Mathematics, EPFL, Switzerland

Acknowlegments: M. Pisaroni, S. Krumscheid, P. Leyland (EPFL), A.L. Haji-Ali (Oxford)
R. Tempone, E. von Schwerin (KAUST)

3rd GAMM AGUQ workshop on Uncertainty Quantification
Dortmund, March 12-14, 2018

EU-FP7 project: *Uncertainty Management for Robust Industrial Design in Aeronautics (UMRIDA)*

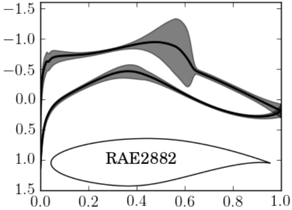
Center for Advanced Modeling Science

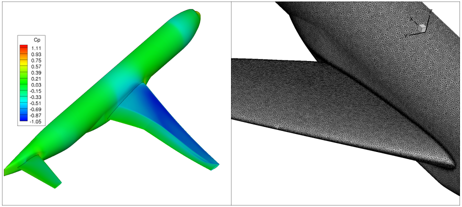
Outline

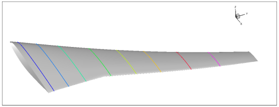
Outline

UQ in aerodynamic design

Compute aerodynamic coeffs. (lift, drag, C_p) and optimize airfoil shape in presence of operational uncertainties (Mach number, angle of attack, ...) and geometrical uncertainties (manufacturing tolerances, icing, fatigue, ...)

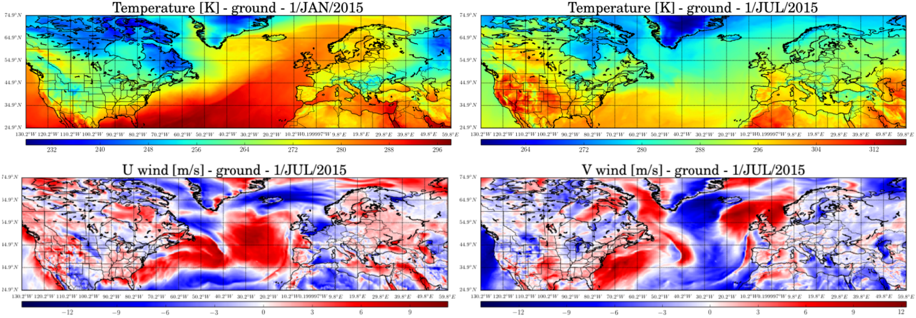






Operational uncertainties

Atmospheric fluctuations with respect to location, time $(T, p, \rho, \boldsymbol{u})$ over long flights



Probabilistic framework: Mach, Reynolds, Angle of Attack, etc. treated as random variables

Geometrical uncertainties

Production: manufacturing, assembly

Temporary factors: deflection, icing

Permanent/degrading factors: impacts, erosion, fouling

Probabilistic Framework: Leading edge radius, thickness, curvature, etc. treated as random variables

Forward Uncertainty propagation

- Random input parameters: y (with given distribution)
- (Complex) Model: $\mathcal{L}_y u = \mathcal{F}$ (e.g. Euler, Navier-Stokes,...) hence $u = u(y)$ is a random solution
- Quantity of interest: $Q = Q(u)$ (random output, e.g. lift, drag, etc.)

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

In practice, u is not accessible. Computational model

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

Forward Uncertainty propagation

- Random input parameters: y (with given distribution)
- (Complex) Model: $\mathcal{L}_y u = \mathcal{F}$ (e.g. Euler, Navier-Stokes,...) hence $u = u(y)$ is a random solution
- Quantity of interest: $Q = Q(u)$ (random output, e.g. lift, drag, etc.)

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

In practice, u is not accessible. Computational model

$$\mathcal{L}_{h,y} u_h = \mathcal{F}_h \quad \Longrightarrow \quad \text{computational output} \quad Q_h = Q(u_h)$$

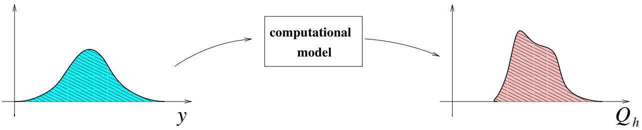
Forward Uncertainty propagation

- Random input parameters: y (with given distribution)
- (Complex) Model: $\mathcal{L}_y u = \mathcal{F}$ (e.g. Euler, Navier-Stokes,...)
  hence $u = u(y)$ is a random solution
- Quantity of interest: $Q = Q(u)$ (random output, e.g. lift, drag, etc.)

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

In practice, u is not accessible. Computational model

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



Monte Carlo method

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

$$\mu_h^{MC} = \frac{1}{M} \sum_{i=1}^{M} Q_h^{(i)} \qquad \text{(biased estimator } \mathbb{E}[\mu_h^{MC}] = \mathbb{E}[Q_h] \neq \mathbb{E}[Q])$$

Mean squared error

$$\mathrm{MSE}(\mu_h^{MC}) := \mathbb{E}[(\mu(Q) - \mu_h^{MC})^2] = \underbrace{(\mathbb{E}[Q - Q_h])^2}_{\text{discret. error}} + \underbrace{\frac{\mathbb{V}\mathrm{ar}[Q_h]}{M}}_{\text{MC error}}$$

Complexity analysis (error versus cost)

Assume:
- $|\mathbb{E}[Q - Q_h]| = \mathcal{O}(h^\alpha)$, $\mathbb{V}\mathrm{ar}[Q_h] = \mathcal{O}(1)$,
- cost to compute each $Q_h^{(i)}$: $C_h = \mathcal{O}(h^{-\gamma})$

Then $\mathrm{MSE} = \mathcal{O}(tol^2) \implies h = \mathcal{O}(tol^{\frac{1}{\alpha}}), \quad M = \mathcal{O}(tol^{-2})$

Total work: $Work(\mu_h^{MC}) = C_h M \lesssim tol^{-\frac{\gamma}{\alpha}} tol^{-2}$

Monte Carlo method

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

$$\mu_h^{MC} = \frac{1}{M} \sum_{i=1}^{M} Q_h^{(i)} \qquad \text{(biased estimator } \mathbb{E}[\mu_h^{MC}] = \mathbb{E}[Q_h] \neq \mathbb{E}[Q])$$

Mean squared error

$$\mathrm{MSE}(\mu_h^{MC}) := \mathbb{E}[(\mu(Q) - \mu_h^{MC})^2] = \underbrace{(\mathbb{E}[Q - Q_h])^2}_{\text{discret. error}} + \underbrace{\frac{\mathbb{V}\mathrm{ar}[Q_h]}{M}}_{\text{MC error}}$$

Complexity analysis (error versus cost)

Assume:
- $|\mathbb{E}[Q - Q_h]| = \mathcal{O}(h^\alpha)$, $\mathbb{V}\mathrm{ar}[Q_h] = \mathcal{O}(1)$,
- cost to compute each $Q_h^{(i)}$: $C_h = \mathcal{O}(h^{-\gamma})$

Then $\mathrm{MSE} = \mathcal{O}(tol^2) \implies h = \mathcal{O}(tol^{\frac{1}{\alpha}}), \quad M = \mathcal{O}(tol^{-2})$

Total work: $Work(\mu_h^{MC}) = C_h M \lesssim tol^{-\frac{\gamma}{\alpha}} tol^{-2}$

Monte Carlo method

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

$$\mu_h^{MC} = \frac{1}{M} \sum_{i=1}^{M} Q_h^{(i)} \qquad \text{(biased estimator } \mathbb{E}[\mu_h^{MC}] = \mathbb{E}[Q_h] \neq \mathbb{E}[Q])$$

Mean squared error

$$\mathrm{MSE}(\mu_h^{MC}) := \mathbb{E}[(\mu(Q) - \mu_h^{MC})^2] = \underbrace{(\mathbb{E}[Q - Q_h])^2}_{\text{discret. error}} + \underbrace{\frac{\mathbb{V}\mathrm{ar}[Q_h]}{M}}_{\text{MC error}}$$

Complexity analysis (error versus cost)

Assume:
- $|\mathbb{E}[Q - Q_h]| = \mathcal{O}(h^\alpha)$, $\mathbb{V}\mathrm{ar}[Q_h] = \mathcal{O}(1)$,
- cost to compute each $Q_h^{(i)}$: $C_h = \mathcal{O}(h^{-\gamma})$

Then $\quad \mathrm{MSE} = \mathcal{O}(tol^2) \quad \Longrightarrow \quad h = \mathcal{O}(tol^{\frac{1}{\alpha}}), \quad M = \mathcal{O}(tol^{-2})$

Total work: $\quad Work(\mu_h^{MC}) = C_h M \lesssim tol^{-\frac{\gamma}{\alpha}} tol^{-2}$

Can we improve on Monte Carlo? Control variate

Let Z be random variable correlated with Q_h, and with known mean.

Idea: Apply MC on $Q_{h,Z} = Q_h - \alpha(Z - \mathbb{E}[Z])$ (notice that $\mathbb{E}[Q_{h,Z}] = \mathbb{E}[Q_h]$)

$$\mu_h^{CV} = \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - \alpha Z^{(i)}) + \alpha\mathbb{E}[Z]$$

$$\mathbb{V}\mathrm{ar}[Q_{h,Z}] = \mathbb{V}\mathrm{ar}[Q_h - \alpha Z] = \mathbb{V}\mathrm{ar}[Q_h] + \alpha^2\mathbb{V}\mathrm{ar}[Z] - 2\alpha\,\mathrm{Cov}(Q_h, Z)$$

For optimal α: $\mathbb{V}\mathrm{ar}[Q_{h,Z}] = \mathbb{V}\mathrm{ar}[Q_h]\left(1 - \frac{\mathrm{Cov}(Q_h,Z)}{\mathbb{V}\mathrm{ar}[Z]}\right) \leq \mathbb{V}\mathrm{ar}[Q_h]$ (always gives variance reduction)

Two ideas for choosing Z

- Use a surrogate model $Z = Q^{surr}$ with numerically optimized α
  $\rightsquigarrow$ *multi-fidelity Monte Carlo* [Peherstorfer, Willcox, Gunzburger, 2016]
- Use coarser discretization e.g. $Z = Q_{2h}$ (usually with $\alpha = 1$)
  $\rightsquigarrow$ *two level Monte Carlo* [Heinrich 1998, Giles 2008, ...]

Can we improve on Monte Carlo? Control variate

Let Z be random variable correlated with Q_h, and with known mean.

Idea: Apply MC on $Q_{h,Z} = Q_h - \alpha(Z - \mathbb{E}[Z])$ (notice that $\mathbb{E}[Q_{h,Z}] = \mathbb{E}[Q_h]$)

$$\mu_h^{CV} = \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - \alpha Z^{(i)}) + \alpha\mathbb{E}[Z]$$

$$\mathbb{V}\text{ar}[Q_{h,Z}] = \mathbb{V}\text{ar}[Q_h - \alpha Z] = \mathbb{V}\text{ar}[Q_h] + \alpha^2\mathbb{V}\text{ar}[Z] - 2\alpha\,\mathrm{Cov}(Q_h, Z)$$

For optimal α: $\mathbb{V}\text{ar}[Q_{h,Z}] = \mathbb{V}\text{ar}[Q_h]\left(1 - \frac{\mathrm{Cov}(Q_h,Z)}{\mathbb{V}\text{ar}[Z]}\right) \leq \mathbb{V}\text{ar}[Q_h]$ (always gives variance reduction)

Two ideas for choosing Z

- Use a surrogate model $Z = Q^{surr}$ with numerically optimized α
  $\rightsquigarrow$ *multi-fidelity Monte Carlo* [Peherstorfer, Willcox, Gunzburger, 2016]
- Use coarser discretization e.g. $Z = Q_{2h}$ (usually with $\alpha = 1$)
  $\rightsquigarrow$ *two level Monte Carlo* [Heinrich 1998, Giles 2008, ...]

Can we improve on Monte Carlo? Control variate

Let Z be random variable correlated with Q_h, and with known mean.

Idea: Apply MC on $Q_{h,Z} = Q_h - \alpha(Z - \mathbb{E}[Z])$ (notice that $\mathbb{E}[Q_{h,Z}] = \mathbb{E}[Q_h]$)

$$\mu_h^{CV} = \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - \alpha Z^{(i)}) + \alpha\mathbb{E}[Z]$$

$$\mathbb{V}\mathrm{ar}[Q_{h,Z}] = \mathbb{V}\mathrm{ar}[Q_h - \alpha Z] = \mathbb{V}\mathrm{ar}[Q_h] + \alpha^2\mathbb{V}\mathrm{ar}[Z] - 2\alpha\,\mathrm{Cov}(Q_h, Z)$$

For optimal α: $\mathbb{V}\mathrm{ar}[Q_{h,Z}] = \mathbb{V}\mathrm{ar}[Q_h]\left(1 - \frac{\mathrm{Cov}(Q_h,Z)}{\mathbb{V}\mathrm{ar}[Z]}\right) \leq \mathbb{V}\mathrm{ar}[Q_h]$ (always gives variance reduction)

Two ideas for choosing Z

- Use a surrogate model $Z = Q^{surr}$ with numerically optimized α
  $\rightsquigarrow$ *multi-fidelity Monte Carlo* [Peherstorfer, Willcox, Gunzburger, 2016]
- Use coarser discretization e.g. $Z = Q_{2h}$ (usually with $\alpha = 1$)
  $\rightsquigarrow$ *two level Monte Carlo* [Heinrich 1998, Giles 2008, ...]

Can we improve on Monte Carlo? Control variate

Problem: $\mathbb{E}[Z]$ not known, in general !

$\rightsquigarrow$ compute it with independent MC with larger sample size (cheaper problem).

From two-level to multilevel:

$$\mu_h^{CV} = \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \mathbb{E}[Q_{2h}]$$

$$\simeq \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \frac{1}{M_2}\sum_{i=1}^{M_2} Q_{2h}^{(i,2)}, \qquad M_2 > M$$

$$\simeq \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \frac{1}{M_2}\sum_{i=1}^{M_2}(Q_{2h}^{(i,2)} - Q_{4h}^{(i,2)}) + \ldots + \frac{1}{M_n}\sum_{i=1}^{M_n} Q_{2^n h}^{(i,n)}$$

Can we improve on Monte Carlo? Control variate

Problem: $\mathbb{E}[Z]$ not known, in general !
$\rightsquigarrow$ compute it with independent MC with larger sample size (cheaper problem).

From two-level to multilevel:

$$\mu_h^{CV} = \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \mathbb{E}[Q_{2h}]$$

$$\simeq \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \frac{1}{M_2}\sum_{i=1}^{M_2} Q_{2h}^{(i,2)}, \qquad M_2 > M$$

$$\simeq \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \frac{1}{M_2}\sum_{i=1}^{M_2}(Q_{2h}^{(i,2)} - Q_{4h}^{(i,2)}) + \ldots + \frac{1}{M_n}\sum_{i=1}^{M_n} Q_{2^n h}^{(i,n)}$$

Can we improve on Monte Carlo? Control variate

Problem: $\mathbb{E}[Z]$ not known, in general !
$\rightsquigarrow$ compute it with independent MC with larger sample size (cheaper problem).

From two-level to multilevel:

$$\mu_h^{CV} = \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \mathbb{E}[Q_{2h}]$$

$$\simeq \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \frac{1}{M_2}\sum_{i=1}^{M_2} Q_{2h}^{(i,2)}, \qquad M_2 > M$$

$$\simeq \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \frac{1}{M_2}\sum_{i=1}^{M_2}(Q_{2h}^{(i,2)} - Q_{4h}^{(i,2)}) + \ldots + \frac{1}{M_n}\sum_{i=1}^{M_n} Q_{2^n h}^{(i,n)}$$

Can we improve on Monte Carlo? Control variate

Problem: $\mathbb{E}[Z]$ not known, in general !
$\rightsquigarrow$ compute it with independent MC with larger sample size (cheaper problem).

From two-level to multilevel:

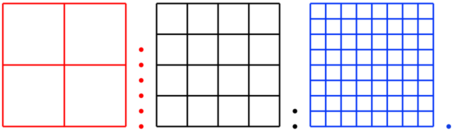
$$\mu_h^{CV} = \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \mathbb{E}[Q_{2h}]$$

$$\simeq \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \frac{1}{M_2}\sum_{i=1}^{M_2} Q_{2h}^{(i,2)}, \qquad M_2 > M$$

$$\simeq \frac{1}{M}\sum_{i=1}^{M}(Q_h^{(i)} - Q_{2h}^{(i)}) + \frac{1}{M_2}\sum_{i=1}^{M_2}(Q_{2h}^{(i,2)} - Q_{4h}^{(i,2)}) + \ldots + \frac{1}{M_n}\sum_{i=1}^{M_n} Q_{2^n h}^{(i,n)}$$

Outline

Multilevel Monte Carlo



$\ell = 0$ $\cdots$ $\ell = L$

- Sequence of refined discretizations

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

- Sequence of sample sizes

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

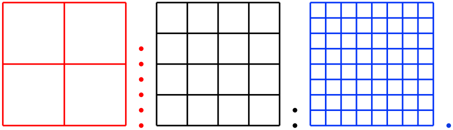
Denoting $Q_\ell = Q_{h_\ell}$, the MLMC estimator is

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

Mean squared error

$$\mathrm{MSE}(\mu_L^{MLMC}) = \underbrace{(\mathbb{E}[Q - Q_L])^2}_{\text{discret. error level } L} + \underbrace{\sum_{\ell=0}^{L} \frac{\mathrm{Var}[Q_\ell - Q_{\ell-1}]}{M_\ell}}_{\text{statistical error}}$$

Multilevel Monte Carlo



$\ell = 0$ $\cdots$ $\ell = L$

- Sequence of refined discretizations

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

- Sequence of sample sizes

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

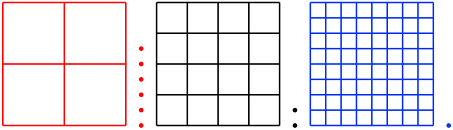
Denoting $Q_\ell = Q_{h_\ell}$, the MLMC estimator is

$$\mu_L^{MLMC} = \frac{1}{M_0}\sum_{i=1}^{M_0} Q_0^{(i,0)} + \frac{1}{M_1}\sum_{i=1}^{M_1}(Q_1^{(i,1)} - Q_0^{(i,1)}) + \ldots + \frac{1}{M_L}\sum_{i=1}^{M_L}(Q_L^{(i,L)} - Q_{L-1}^{(i,L)})$$

Mean squared error

$$\mathrm{MSE}(\mu_L^{MLMC}) = \underbrace{(\mathbb{E}[Q - Q_L])^2}_{\text{discret. error level } L} + \underbrace{\sum_{\ell=0}^{L} \frac{\mathrm{Var}[Q_\ell - Q_{\ell-1}]}{M_\ell}}_{\text{statistical error}}$$

Multilevel Monte Carlo



$\ell = 0$ $\cdots$ $\ell = L$

- Sequence of refined discretizations

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

- Sequence of sample sizes

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

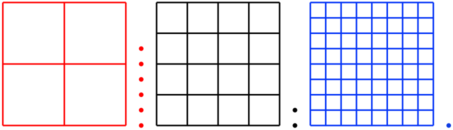
Denoting $Q_\ell = Q_{h_\ell}$, the MLMC estimator is

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

Mean squared error

$$\mathrm{MSE}(\mu_L^{MLMC}) = \underbrace{(\mathbb{E}[Q - Q_L])^2}_{\text{discret. error level } L} + \underbrace{\sum_{\ell=0}^{L} \frac{\mathrm{Var}[Q_\ell - Q_{\ell-1}]}{M_\ell}}_{\text{statistical error}}$$

Multilevel Monte Carlo



$\ell = 0$ $\cdots$ $\ell = L$

- Sequence of refined discretizations

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

- Sequence of sample sizes

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

Denoting $Q_\ell = Q_{h_\ell}$, the MLMC estimator is

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

Mean squared error

$$\mathrm{MSE}(\mu_L^{MLMC}) = \underbrace{(\mathbb{E}[Q - Q_L])^2}_{\text{discret. error level } L} + \underbrace{\sum_{\ell=0}^{L} \frac{\mathrm{Var}[Q_\ell - Q_{\ell-1}]}{M_\ell}}_{\text{statistical error}}$$

Multilevel Monte Carlo

- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$ (variance of differences)
- $C_\ell =$ cost of computing each $\Delta Q_\ell^{(i,\ell)} = Q_\ell^{(i,\ell)} - Q_{\ell-1}^{(i,\ell)}$

Optimal sample sizes M_ℓ: [Giles 2008] minimize $W = \sum_{\ell=0}^{L} C_\ell M_\ell$ s.t. $\mathrm{MSE} \simeq tol^2$

$$M_\ell = \left\lceil tol^{-2} \sqrt{\frac{V_\ell}{C_\ell}} \left(\sum_{k=0}^{L} \sqrt{C_k V_k} \right) \right\rceil$$

Complexity analysis for $h_\ell = h_0 s^{-\ell}$: [Giles 2008, Cliffe-Giles-Scheichl-Teckentrup 2011]

Assume

- $|\mathbb{E}[Q - Q_\ell]| = \mathcal{O}(h_\ell^\alpha)$,
- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] = O(h_\ell^\beta)$, $\quad (\beta = 2\alpha$ for smooth problems/noise)
- $C_\ell = \mathcal{O}(h_\ell^{-\gamma})$, $\quad 2\alpha \geq \min\{\beta, \gamma\}$

Then, choosing $L = \mathcal{O}(tol^{\frac{1}{\alpha}})$ and M_ℓ as above gives $\mathrm{MSE}(\mu_L^{MLMC}) \leq tol^2$ and

$$Work(\mu_L^{MLMC}) = \sum_{\ell=0}^{L} C_\ell M_\ell \lesssim \begin{cases} tol^{-2}, & \beta > \gamma \\ tol^{-2}(\log tol)^2, & \beta = \gamma \\ tol^{-2-\frac{\gamma-\beta}{\alpha}}, & \beta < \gamma \end{cases}$$

Multilevel Monte Carlo

- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$ (variance of differences)
- C_ℓ = cost of computing each $\Delta Q_\ell^{(i,\ell)} = Q_\ell^{(i,\ell)} - Q_{\ell-1}^{(i,\ell)}$

Optimal sample sizes M_ℓ: [Giles 2008] minimize $W = \sum_{\ell=0}^{L} C_\ell M_\ell$ s.t. $\mathrm{MSE} \simeq tol^2$

$$M_\ell = \left\lceil tol^{-2}\sqrt{\frac{V_\ell}{C_\ell}}\left(\sum_{k=0}^{L}\sqrt{C_k V_k}\right)\right\rceil$$

Complexity analysis for $h_\ell = h_0 s^{-\ell}$: [Giles 2008, Cliffe-Giles-Scheichl-Teckentrup 2011]

Assume

- $|\mathbb{E}[Q - Q_\ell]| = \mathcal{O}(h_\ell^\alpha)$,
- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] = O(h_\ell^\beta)$, ($\beta = 2\alpha$ for smooth problems/noise)
- $C_\ell = \mathcal{O}(h_\ell^{-\gamma})$, $2\alpha \geq \min\{\beta, \gamma\}$

Then, choosing $L = \mathcal{O}(tol^{\frac{1}{\alpha}})$ and M_ℓ as above gives $\mathrm{MSE}(\mu_L^{MLMC}) \leq tol^2$ and

$$Work(\mu_L^{MLMC}) = \sum_{\ell=0}^{L} C_\ell M_\ell \lesssim \begin{cases} tol^{-2}, & \beta > \gamma \\ tol^{-2}(\log tol)^2, & \beta = \gamma \\ tol^{-2-\frac{\gamma-\beta}{\alpha}}, & \beta < \gamma \end{cases}$$

Multilevel Monte Carlo

- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}]$ (variance of differences)
- $C_\ell =$ cost of computing each $\Delta Q_\ell^{(i,\ell)} = Q_\ell^{(i,\ell)} - Q_{\ell-1}^{(i,\ell)}$

Optimal sample sizes M_ℓ: [Giles 2008] minimize $W = \sum_{\ell=0}^{L} C_\ell M_\ell$ s.t. $\mathrm{MSE} \simeq tol^2$

$$M_\ell = \left\lceil tol^{-2} \sqrt{\frac{V_\ell}{C_\ell}} \left(\sum_{k=0}^{L} \sqrt{C_k V_k} \right) \right\rceil$$

Complexity analysis for $h_\ell = h_0 s^{-\ell}$: [Giles 2008, Cliffe-Giles-Scheichl-Teckentrup 2011]

Assume

- $|\mathbb{E}[Q - Q_\ell]| = \mathcal{O}(h_\ell^\alpha)$,
- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] = O(h_\ell^\beta)$, $\qquad (\beta = 2\alpha$ for smooth problems/noise$)$
- $C_\ell = \mathcal{O}(h_\ell^{-\gamma})$, $\qquad 2\alpha \geq \min\{\beta, \gamma\}$

Then, choosing $L = \mathcal{O}(tol^{\frac{1}{\alpha}})$ and M_ℓ as above gives $\mathrm{MSE}(\mu_L^{MLMC}) \leq tol^2$ and

$$Work(\mu_L^{MLMC}) = \sum_{\ell=0}^{L} C_\ell M_\ell \lesssim \begin{cases} tol^{-2}, & \beta > \gamma \\ tol^{-2}(\log tol)^2, & \beta = \gamma \\ tol^{-2-\frac{\gamma-\beta}{\alpha}}, & \beta < \gamma \end{cases}$$

Multilevel Monte Carlo – practical aspects

Remark: MC complexity always improved for optimal choice of M_ℓ.
For $\beta = 2\alpha$ we get either $\mathcal{O}(tol^{-2})$ (up to log terms) or $\mathcal{O}(tol^{-\frac{\gamma}{\alpha}})$.

To achieve improved complexity, one needs to

- estimate error decay $|\mathbb{E}[Q - Q_\ell]|$: $\rightsquigarrow$ needed to determine optimal L
- estimate variance decay V_ℓ: $\rightsquigarrow$ needed to determine optimal $\{M_\ell\}_{\ell=0}^{L}$

$|\mathbb{E}[Q - Q_\ell]|$ can be estimated as $|\mu_\ell^{MC} - \mu_{\ell-1}^{MC}|$ based on a pilot run
V_ℓ can be estimated by sample variance estimator based on pilot runs

Problem: on the finest levels we should run only very few simulations.
Cost for estimation of V_L might dominate the overall cost of the MLMC algorithm.

Idea: use adaptive algorithms: extrapolate information from previous levels and correct it when samples become available.

Multilevel Monte Carlo – practical aspects

Remark: MC complexity always improved for optimal choice of M_ℓ.
For $\beta = 2\alpha$ we get either $\mathcal{O}(tol^{-2})$ (up to log terms) or $\mathcal{O}(tol^{-\frac{\gamma}{\alpha}})$.

To achieve improved complexity, one needs to

- estimate error decay $|\mathbb{E}[Q - Q_\ell]|$: $\rightsquigarrow$ needed to determine optimal L
- estimate variance decay V_ℓ: $\rightsquigarrow$ needed to determine optimal $\{M_\ell\}_{\ell=0}^{L}$

$|\mathbb{E}[Q - Q_\ell]|$ can be estimated as $|\mu_\ell^{MC} - \mu_{\ell-1}^{MC}|$ based on a pilot run
V_ℓ can be estimated by sample variance estimator based on pilot runs

Problem: on the finest levels we should run only very few simulations.
Cost for estimation of V_L might dominate the overall cost of the MLMC algorithm.

Idea: use adaptive algorithms: extrapolate information from previous levels and correct it when samples become available.

Multilevel Monte Carlo – practical aspects

Remark: MC complexity always improved for optimal choice of M_ℓ.
For $\beta = 2\alpha$ we get either $\mathcal{O}(tol^{-2})$ (up to log terms) or $\mathcal{O}(tol^{-\frac{\gamma}{\alpha}})$.

To achieve improved complexity, one needs to

- estimate error decay $|\mathbb{E}[Q - Q_\ell]|$: $\rightsquigarrow$ needed to determine optimal L
- estimate variance decay V_ℓ: $\rightsquigarrow$ needed to determine optimal $\{M_\ell\}_{\ell=0}^{L}$

$|\mathbb{E}[Q - Q_\ell]|$ can be estimated as $|\mu_\ell^{MC} - \mu_{\ell-1}^{MC}|$ based on a pilot run
V_ℓ can be estimated by sample variance estimator based on pilot runs

Problem: on the finest levels we should run only very few simulations.
Cost for estimation of V_L might dominate the overall cost of the MLMC algorithm.

Idea: use adaptive algorithms: extrapolate information from previous levels and correct it when samples become available.

Multilevel Monte Carlo – practical aspects

Remark: MC complexity always improved for optimal choice of M_ℓ.
For $\beta = 2\alpha$ we get either $\mathcal{O}(tol^{-2})$ (up to log terms) or $\mathcal{O}(tol^{-\frac{\gamma}{\alpha}})$.

To achieve improved complexity, one needs to

- estimate error decay $|\mathbb{E}[Q - Q_\ell]|$: $\rightsquigarrow$ needed to determine optimal L
- estimate variance decay V_ℓ: $\rightsquigarrow$ needed to determine optimal $\{M_\ell\}_{\ell=0}^{L}$

$|\mathbb{E}[Q - Q_\ell]|$ can be estimated as $|\mu_\ell^{MC} - \mu_{\ell-1}^{MC}|$ based on a pilot run
V_ℓ can be estimated by sample variance estimator based on pilot runs

Problem: on the finest levels we should run only very few simulations.
Cost for estimation of V_L might dominate the overall cost of the MLMC algorithm.

Idea: use adaptive algorithms: extrapolate information from previous levels and correct it when samples become available.

Multilevel Monte Carlo – practical aspects

Remark: MC complexity always improved for optimal choice of M_ℓ.
For $\beta = 2\alpha$ we get either $\mathcal{O}(tol^{-2})$ (up to log terms) or $\mathcal{O}(tol^{-\frac{\gamma}{\alpha}})$.

To achieve improved complexity, one needs to

- estimate error decay $|\mathbb{E}[Q - Q_\ell]|$: $\rightsquigarrow$ needed to determine optimal L
- estimate variance decay V_ℓ: $\rightsquigarrow$ needed to determine optimal $\{M_\ell\}_{\ell=0}^{L}$

$|\mathbb{E}[Q - Q_\ell]|$ can be estimated as $|\mu_\ell^{MC} - \mu_{\ell-1}^{MC}|$ based on a pilot run
V_ℓ can be estimated by sample variance estimator based on pilot runs

Problem: on the finest levels we should run only very few simulations.
Cost for estimation of V_L might dominate the overall cost of the MLMC algorithm.

Idea: use adaptive algorithms: extrapolate information from previous levels and correct it when samples become available.

Continuation Multilevel Monte Carlo

[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_ℓ and $|\mathbb{E}[Q - Q_\ell]|$.

Estimator $\hat{V}_\ell$ of $V_\ell = \mathbb{V}\mathrm{ar}[\Delta Q_\ell]$ at iteration j: MAP Bayesian estimator

- we make the ansatz $\Delta Q_\ell \sim N(\mu_\ell, V_\ell)$
- 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)
$M_\ell \to \infty$	$\hat{V}_\ell \approx V_\ell^{MC}$	(sample variance)

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

Continuation Multilevel Monte Carlo

[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_ℓ and $|\mathbb{E}[Q - Q_\ell]|$.

Estimator $\hat{V}_\ell$ of $V_\ell = \mathbb{V}\mathrm{ar}[\Delta Q_\ell]$ at iteration j: MAP Bayesian estimator

- we make the ansatz $\Delta Q_\ell \sim N(\mu_\ell, V_\ell)$
- 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

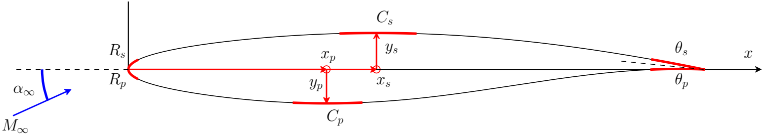
Effectively, we have

$M_\ell = 0$	$\hat{V}_\ell = V_\ell^{model}$	(prior model)
$M_\ell \to \infty$	$\hat{V}_\ell \approx V_\ell^{MC}$	(sample variance)

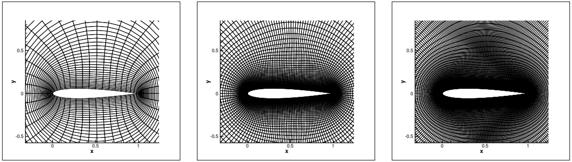
$\hat{V}_\ell$ is then used to determine the sample sizes M_ℓ for the next iteration.

Computation of pressure coefficient for NACA 0012 / NASA SC(2)-0012 airfoils

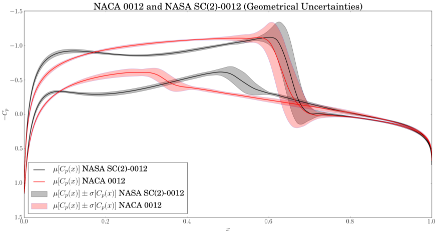
	Name	Nominal value	Uncertainty
Operational	T_∞	$T_n = 288.15$ [K]	$\mathcal{TN}(T_n, 2\%, 110\%, 90\%)$
	p_∞	$p_n = 101325$ [N/m^2]	$\mathcal{TN}(p_n, 2\%, 110\%, 90\%)$
	α	$\alpha_n = 1.25°$	$\mathcal{TN}(\alpha_n, 1\%, 110\%, 90\%)$
	M	$M_n = 0.8$	$\mathcal{TN}(M_n, 2\%, 110\%, 90\%)$
Geometrical	R_p	0.01458398	$\mathcal{TN}(R_{P_n}, 2.5\%, 110\%, 90\%)$
	R_S	0.01458398	$\mathcal{TN}(R_{S_n}, 2.5\%, 110\%, 90\%)$
	X_P	0.30049047	$\mathcal{TN}(X_{P_n}, 2.5\%, 110\%, 90\%)$
	X_S	0.30049047	$\mathcal{TN}(X_{S_n}, 2.5\%, 110\%, 90\%)$
	Y_P	-0.05994286	$\mathcal{TN}(Y_{P_n}, 2.5\%, 110\%, 90\%)$
	Y_S	0.05994286	$\mathcal{TN}(Y_{S_n}, 2.5\%, 110\%, 90\%)$
	C_P	0.44213792	$\mathcal{TN}(C_{P_n}, 2.5\%, 110\%, 90\%)$
	C_S	-0.44213792	$\mathcal{TN}(C_{S_n}, 2.5\%, 110\%, 90\%)$
	θ_P	8.3763395	$\mathcal{TN}(\theta_{P_n}, 2.5\%, 110\%, 90\%)$
	θ_S	-8.3763395	$\mathcal{TN}(\theta_{S_n}, 2.5\%, 110\%, 90\%)$

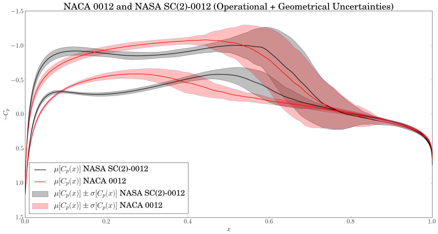


Computation of pressure coefficient for NACA 0012 / NASA SC(2)-0012 airfoils



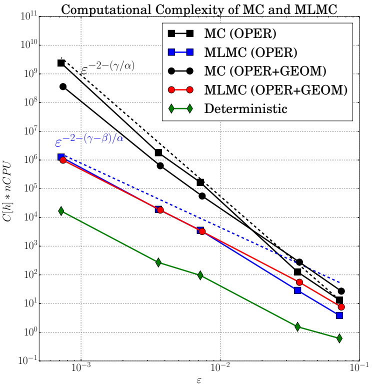
LEVEL	Airfoil nodes	Cells	Avg. Real Computational Time [s] (CPU)
$L0$	41	6943	12.4 (32)
$L1$	81	11115	20.9 (38)
$L2$	161	19385	26.9 (44)
$L3$	321	36251	71.1 (50)
$L4$	641	71477	231.15 (56)
$L5$	1281	145005	422.0 (64)

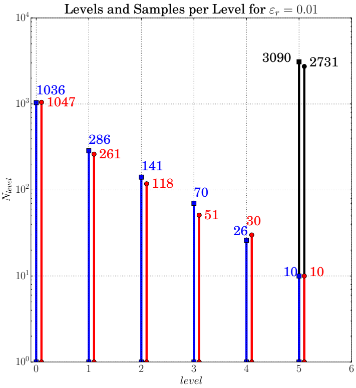




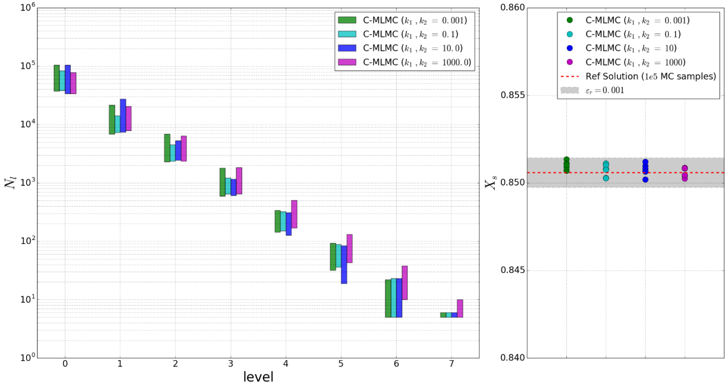
Inviscid model (Euler); SU2 solver (Stanford) [Pisaroni-Leyland-N., AIAA Aviation, 2016]

MLMC vs MC for aerodynamic inviscid problems





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? [Bierig-Chernov 2015-2016] use biased central moments estimators.

Alternatively, use *h*-statistics [Pisaroni-Krumscheid-N. 2017]. Given iid sample $\vec{Q}_M = \{Q^{(1)}, \dots, Q^{(M)}\}$,

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

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

with $(\vec{Q}_{\ell,M_\ell}, \vec{Q}_{\ell-1,M_\ell})$ generated with the same noise (highly correlated)

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

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

Same "formal" structure as for expectation, but now we need to estimate $|\mu_p(Q) - \mu_p(Q_\ell)|$ and $V_{\ell,p}$ to tune the MLMC algorithm

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? [Bierig-Chernov 2015-2016] use biased central moments estimators.

Alternatively, use *h*-statistics [Pisaroni-Krumscheid-N. 2017]. Given iid sample $\vec{Q}_M = \{Q^{(1)}, \dots, Q^{(M)}\}$,

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

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

with $(\vec{Q}_{\ell,M_\ell}, \vec{Q}_{\ell-1,M_\ell})$ generated with the same noise (highly correlated)

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

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

Same "formal" structure as for expectation, but now we need to estimate $|\mu_p(Q) - \mu_p(Q_\ell)|$ and $V_{\ell,p}$ to tune the MLMC algorithm

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? [Bierig-Chernov 2015-2016] use biased central moments estimators.

Alternatively, use *h*-statistics [Pisaroni-Krumscheid-N. 2017]. Given iid sample $\vec{Q}_M = \{Q^{(1)}, \dots, Q^{(M)}\}$,

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

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

with $(\vec{Q}_{\ell,M_\ell}, \vec{Q}_{\ell-1,M_\ell})$ generated with the same noise (highly correlated)

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

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

Same "formal" structure as for expectation, but now we need to estimate $|\mu_p(Q) - \mu_p(Q_\ell)|$ and $V_{\ell,p}$ to tune the MLMC algorithm

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? [Bierig-Chernov 2015-2016] use biased central moments estimators.

Alternatively, use *h*-statistics [Pisaroni-Krumscheid-N. 2017]. Given iid sample $\vec{Q}_M = \{Q^{(1)}, \dots, Q^{(M)}\}$,

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

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

with $(\vec{Q}_{\ell,M_\ell}, \vec{Q}_{\ell-1,M_\ell})$ generated with the same noise (highly correlated)

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

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

Same "formal" structure as for expectation, but now we need to estimate $|\mu_p(Q) - \mu_p(Q_\ell)|$ and $V_{\ell,p}$ to tune the MLMC algorithm

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? [Bierig-Chernov 2015-2016] use biased central moments estimators.

Alternatively, use *h*-statistics [Pisaroni-Krumscheid-N. 2017]. Given iid sample $\vec{Q}_M = \{Q^{(1)}, \dots, Q^{(M)}\}$,

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

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

with $(\vec{Q}_{\ell,M_\ell}, \vec{Q}_{\ell-1,M_\ell})$ generated with the same noise (highly correlated)

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

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

Same "formal" structure as for expectation, but now we need to estimate $|\mu_p(Q) - \mu_p(Q_\ell)|$ and $V_{\ell,p}$ to tune the MLMC algorithm

Beyond expectations: computation of central moments

Complexity result for $h_\ell = h_0 s^{-\ell}$

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

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

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

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

Beyond expectations: computation of central moments

Technical difficulty: how to estimate the variances $V_{\ell,p}$

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: computation of central moments

Technical difficulty: how to estimate the variances $V_{\ell,p}$

Define $\vec{X}^{+}_{\ell,M_\ell} = \vec{Q}_{\ell,M_\ell} + \vec{Q}_{\ell-1,M_\ell}, \quad \vec{X}^{-}_{\ell,M_\ell} = \vec{Q}_{\ell,M_\ell} - \vec{Q}_{\ell-1,M_\ell}$
$\Delta_\ell h_p = h_p(\vec{Q}_{\ell,M_\ell}) - h_p(\vec{Q}_{\ell-1,M_\ell})$ 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: computation of central moments

Technical difficulty: how to estimate the variances $V_{\ell,p}$

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: char. function, CDF, and more

Some derived quantities can be written as parametric expectations

Example 1: Characteristic function of Q

$$\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)], \qquad \phi(\theta, Q) = e^{i\theta Q}$$

$\rightsquigarrow$ we can compute $\Phi(\theta_j)$ by MLMC on a set of points θ_j.

Example 2: CDF of Q

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

Problem: $\phi(\theta, Q)$ is not smooth ! When applying MLMC, 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 MLMC.

Remedies:

- [Giles-Nagapetyan-Ritter 2015] 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 (see Alexey's talk)
- [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: char. function, CDF, and more

Some derived quantities can be written as parametric expectations

Example 1: Characteristic function of Q

$$\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)], \qquad \phi(\theta, Q) = e^{i\theta Q}$$

$\rightsquigarrow$ we can compute $\Phi(\theta_j)$ by MLMC on a set of points θ_j.

Example 2: CDF of Q

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

Problem: $\phi(\theta, Q)$ is not smooth ! When applying MLMC, 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 MLMC.

Remedies:

- [Giles-Nagapetyan-Ritter 2015] 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 (see Alexey's talk)
- [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: char. function, CDF, and more

Some derived quantities can be written as parametric expectations

Example 1: Characteristic function of Q

$$\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)], \qquad \phi(\theta, Q) = e^{i\theta Q}$$

$\rightsquigarrow$ we can compute $\Phi(\theta_j)$ by MLMC on a set of points θ_j.

Example 2: CDF of Q

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

Problem: $\phi(\theta, Q)$ is not smooth ! When applying MLMC, 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 MLMC.

Remedies:

- [Giles-Nagapetyan-Ritter 2015] 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 (see Alexey's talk)
- [Krumscheid-N. 2017] anti-derivative approach: $F(\theta) = \Phi'(\theta)$ with $\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)]$ and $\phi(\theta, \cdot)$ Lipschitz continuous.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Beyond expectations: char. function, CDF, and more

Some derived quantities can be written as parametric expectations

Example 1: Characteristic function of Q

$$\Phi(\theta) = \mathbb{E}[\phi(\theta, Q)], \qquad \phi(\theta, Q) = e^{i\theta Q}$$

$\rightsquigarrow$ we can compute $\Phi(\theta_j)$ by MLMC on a set of points θ_j.

Example 2: CDF of Q

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

Problem: $\phi(\theta, Q)$ is not smooth ! When applying MLMC, 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 MLMC.

Remedies:

- [Giles-Nagapetyan-Ritter 2015] 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 (see Alexey's talk)
- [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 any $\tau \in (0,1)$ define

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

Then

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

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 any $\tau \in (0,1)$ define

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

Then

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

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

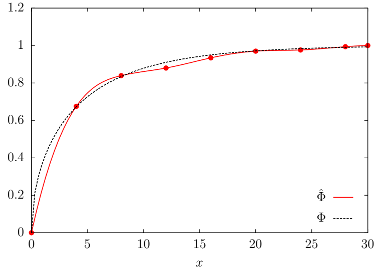
Moreover, from the approximation of Φ_τ and its derivatives we can get for free

- 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.



Interpolation approach:

- introduce a grid $\vec{\xi} = \{\xi_1, \ldots, \xi_n\} \subset \Theta$
- compute $\Phi_L^{MLMC}(\xi_j)$, $j = 1, \ldots, n$ by MLMC (same sample of Q_ℓ for every ξ_j)
- Interpolate values $\Phi_L^{MLMC}(\vec{\xi}) = \{\Phi_L^{MLMC}(\xi_j)\}_{j=1}^n$

$$\hat{\Phi}_L = \mathcal{I}_n(\Phi_L^{MLMC}(\vec{\xi}))$$

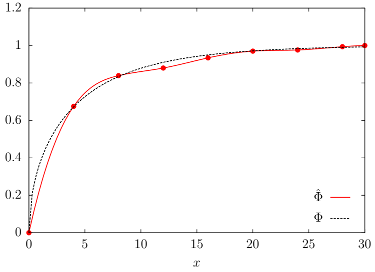
e.g. by spline or polynomial interpolation

Assumptions on $\mathcal{I}_n$ (valid for spline interpolation)

- $\|f - \mathcal{I}_n(f(\vec{\xi}))\|_{L^\infty(\Theta)} \leq c_1 n^{k+1}, \quad$ if $f \in C^{k+1}(\bar{\Theta})$
- $\|\mathcal{I}_n \vec{x}\|_{L^\infty(\Theta)} \leq c_2 \|\vec{x}\|_{\ell^\infty}, \quad \forall \vec{x} \in \mathbb{R}^n$
- $Cost(\mathcal{I}_n(\vec{x})) \leq c_3 n$

Computing parametric expectations by MLMC

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



Interpolation approach:

- introduce a grid $\vec{\xi} = \{\xi_1, \dots, \xi_n\} \subset \Theta$
- compute $\Phi_L^{MLMC}(\xi_j)$, $j = 1, \dots, n$ by MLMC (same sample of Q_ℓ for every ξ_j)
- Interpolate values $\Phi_L^{MLMC}(\vec{\xi}) = \{\Phi_L^{MLMC}(\xi_j)\}_{j=1}^n$

$$\hat{\Phi}_L = \mathcal{I}_n(\Phi_L^{MLMC}(\vec{\xi}))$$

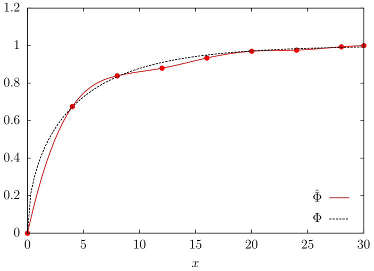
e.g. by spline or polynomial interpolation

Assumptions on $\mathcal{I}_n$ (valid for spline interpolation)

- $\|f - \mathcal{I}_n(f(\vec{\xi}))\|_{L^\infty(\Theta)} \le c_1 n^{k+1}, \quad$ if $f \in C^{k+1}(\bar{\Theta})$
- $\|\mathcal{I}_n \vec{x}\|_{L^\infty(\Theta)} \le c_2 \|\vec{x}\|_{\ell^\infty}, \quad \forall \vec{x} \in \mathbb{R}^n$
- $Cost(\mathcal{I}_n(\vec{x})) \le c_3 n$

Computing parametric expectations by MLMC

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



Interpolation approach:

- introduce a grid $\vec{\xi} = \{\xi_1, \dots, \xi_n\} \subset \Theta$
- compute $\Phi_L^{MLMC}(\xi_j)$, $j = 1, \dots, n$ by MLMC (same sample of Q_ℓ for every ξ_j)
- Interpolate values $\Phi_L^{MLMC}(\vec{\xi}) = \{\Phi_L^{MLMC}(\xi_j)\}_{j=1}^n$

$$\hat{\Phi}_L = \mathcal{I}_n(\Phi_L^{MLMC}(\vec{\xi}))$$

e.g. by spline or polynomial interpolation

Assumptions on $\mathcal{I}_n$ (valid for spline interpolation)

- $\|f - \mathcal{I}_n(f(\vec{\xi}))\|_{L^\infty(\Theta)} \le c_1 n^{k+1}, \quad$ if $f \in C^{k+1}(\bar{\Theta})$
- $\|\mathcal{I}_n \vec{x}\|_{L^\infty(\Theta)} \le c_2 \|\vec{x}\|_{\ell^\infty}, \quad \forall \vec{x} \in \mathbb{R}^n$
- $Cost(\mathcal{I}_n(\vec{x})) \le c_3 n$

Error splitting

Define the mean squared error: $\mathrm{MSE}(\hat{\Phi}_L) = \mathbb{E}[\|\Phi - \hat{\Phi}_L\|^2_{L^\infty(\Theta)}]$

Notation: for $\vec{x} \in \mathbb{R}^n$ define $\mathbb{V}\mathrm{ar}[\vec{x}] = \mathbb{E}[\|\vec{x} - \mathbb{E}[\vec{x}]\|^2_{\ell^\infty}]$

Useful result: for $\vec{x}^{(1)}, \dots, \vec{x}^{(k)} \in \mathbb{R}^n$ independent,

$$\mathbb{V}\mathrm{ar}[\sum_{i=1}^{k} \vec{x}^{(i)}] \leq c \log(n) \sum_{i=1}^{k} \mathbb{V}\mathrm{ar}[\vec{x}^{(i)}]$$

Error splitting

$$\mathrm{MSE}(\hat{\Phi}_L) \leq 3\|\Phi - \mathcal{I}_n\Phi\|^2_\infty + 3\|\mathcal{I}_n\Phi - \mathcal{I}_n\Phi_L\|^2_\infty + 3\mathbb{E}[\|\mathcal{I}_n\Phi_L - \mathcal{I}_n\Phi_L^{MLMC}\|^2_\infty]$$

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

with $V_\ell = \mathbb{V}\mathrm{ar}[\phi(\vec{\xi}, Q_\ell) - \phi(\vec{\xi}, Q_{\ell-1})]$. All terms can be estimated in practice. Optimization of MLMC based on estimators $\hat{V}_\ell$.

[Pisaroni-Krumscheid-N. in preparation]

Error splitting

Define the mean squared error: $\mathrm{MSE}(\hat{\Phi}_L) = \mathbb{E}[\|\Phi - \hat{\Phi}_L\|^2_{L^\infty(\Theta)}]$

Notation: for $\vec{x} \in \mathbb{R}^n$ define $\mathbb{V}\mathrm{ar}[\vec{x}] = \mathbb{E}[\|\vec{x} - \mathbb{E}[\vec{x}]\|^2_{\ell^\infty}]$

Useful result: for $\vec{x}^{(1)}, \dots, \vec{x}^{(k)} \in \mathbb{R}^n$ independent,

$$\mathbb{V}\mathrm{ar}[\sum_{i=1}^{k} \vec{x}^{(i)}] \leq c \log(n) \sum_{i=1}^{k} \mathbb{V}\mathrm{ar}[\vec{x}^{(i)}]$$

Error splitting

$$\begin{aligned}\mathrm{MSE}(\hat{\Phi}_L) &\leq 3\|\Phi - \mathcal{I}_n\Phi\|^2_\infty + 3\|\mathcal{I}_n\Phi - \mathcal{I}_n\Phi_L\|^2_\infty + 3\mathbb{E}[\|\mathcal{I}_n\Phi_L - \mathcal{I}_n\Phi_L^{MLMC}\|^2_\infty] \\ &\lesssim \underbrace{\|\Phi - \mathcal{I}_n\Phi(\vec{\xi})\|^2_\infty}_{\text{interp. error}} + \underbrace{\|\Phi(\vec{\xi}) - \Phi_L(\vec{\xi})\|^2_\infty}_{\text{discret. error}} + \log(n) \underbrace{\sum_{\ell=0}^{L} \frac{V_\ell}{M_\ell}}_{\text{statistical error}}\end{aligned}$$

with $V_\ell = \mathbb{V}\mathrm{ar}[\phi(\vec{\xi}, Q_\ell) - \phi(\vec{\xi}, Q_{\ell-1})]$. All terms can be estimated in practice. Optimization of MLMC based on estimators $\hat{V}_\ell$.

[Pisaroni-Krumscheid-N. in preparation]

Complexity analysis

Complexity result for $h_\ell = h_0 s^{-\ell}$ [Krumscheid-N. 2017]

Assume

- $\|\Phi - \Phi_\ell\|_{L^\infty(\Theta)} \leq c_1 {h_\ell}^\alpha$,
- $\mathbb{E}\Big(\|\phi(\cdot, Q_\ell) - \phi(\cdot, Q_{\ell-1})\|^2_{L^\infty(\Theta)}\Big) \leq c_2 {h_\ell}^\beta$,
- cost to simulate one realization of $\phi(\theta, Q_\ell) \leq c_3 {h_\ell}^{-\gamma}$.

If $\Phi \in C^{k+1}(\Theta)$, there exists an estimator $\hat{\Phi}_L$ s.t. $\mathrm{MSE}(\hat{\Phi}_L) = \mathcal{O}(tol^2)$ and

$$W(\hat{\Phi}_L) \lesssim tol^{-\left(2+\frac{1}{k+1}\right)} |\log(tol)| + |\log(tol)| \begin{cases} tol^{-2}, & \text{if } \beta > \gamma, \\ tol^{-2}|\log(tol)|^2, & \text{if } \beta = \gamma, \\ tol^{-\left(2+\frac{\gamma-\beta}{\alpha}\right)}, & \text{if } \beta < \gamma, \end{cases}$$

The first term accounts for the cost of computing the spline interpolation. This is often negligible for heavy computational models. It can be removed by taking $n = n_\ell$ (different spline interpolant on each level).

Neglecting the first term, the complexity is essentially the same as for simple expectations, up to an extra log factor.

Complexity analysis

Complexity result for $h_\ell = h_0 s^{-\ell}$ [Krumscheid-N. 2017]

Assume

- $\|\Phi - \Phi_\ell\|_{L^\infty(\Theta)} \leq c_1 {h_\ell}^{\alpha}$,
- $\mathbb{E}\Big(\|\phi(\cdot, Q_\ell) - \phi(\cdot, Q_{\ell-1})\|^2_{L^\infty(\Theta)}\Big) \leq c_2 {h_\ell}^{\beta}$,
- cost to simulate one realization of $\phi(\theta, Q_\ell) \leq c_3 {h_\ell}^{-\gamma}$.

If $\Phi \in C^{k+1}(\Theta)$, there exists an estimator $\hat{\Phi}_L$ s.t. $\mathrm{MSE}(\hat{\Phi}_L) = \mathcal{O}(tol^2)$ and

$$W(\hat{\Phi}_L) \lesssim tol^{-\left(2+\frac{1}{k+1}\right)} |\log(tol)| + |\log(tol)| \begin{cases} tol^{-2}, & \text{if } \beta > \gamma, \\ tol^{-2} |\log(tol)|^2, & \text{if } \beta = \gamma, \\ tol^{-\left(2+\frac{\gamma-\beta}{\alpha}\right)}, & \text{if } \beta < \gamma, \end{cases}$$

The first term accounts for the cost of computing the spline interpolation. This is often negligible for heavy computational models. It can be removed by taking $n = n_\ell$ (different spline interpolant on each level).

Neglecting the first term, the complexity is essentially the same as for simple expectations, up to an extra log factor.

Complexity result for derivatives [Krumscheid-N. 2017]

If $\Phi \in C^{2k+2}(\Theta)$ and $m \leq 2k+1$, there exists an estimator $\hat{\Phi}_L$ s.t. $\mathbb{E}[\| \frac{d^m}{d\theta^m}\Phi - \frac{d^m}{d\theta^m}\hat{\Phi}_L\|_\infty^2] = \mathcal{O}(tol^2)$ and

$$W(\hat{\Phi}_L) \lesssim |\log(tol)| \begin{cases} tol^{-2\frac{2k+2}{2k+2-m}}, & \text{if } \beta > \gamma, \\ tol^{-2\frac{2k+2}{2k+2-m}} |\log(tol)|^2, & \text{if } \beta = \gamma, \\ tol^{-\left(2+\frac{\gamma-\beta}{\alpha}\right)\frac{2k+2}{2k+2-m}}, & \text{if } \beta < \gamma, \end{cases}$$

(neglecting the cost of interpolation)

This result applies to the approximation of CDF, quantiles and CVaR with $m = 1$ and PDF with $m = 2$.

An example: the characteristic function

- An SDE model to describe a European call option, where the asset follows

$$dS = rS\,dt + \sigma S\,dW\ , \quad S(0) = S_0\ ,$$

- Quantity of interest is the discounted "payoff": $Q := e^{-rT}\max\big(S(T) - K, 0\big)$
- Approximate **characteristic function** of Q:

$$\Phi(\theta) = \mathbb{E}\big(\cos(\theta Q)\big) + \mathrm{i}\,\mathbb{E}\big(\sin(\theta Q)\big) \equiv \Phi_1(\theta) + \mathrm{i}\,\Phi_2(\theta)\ ,$$

- Milstein scheme with $h_\ell = 2^{-\ell}T$; $\Theta = [-1, 1]$, $r = \frac{1}{20}$, $\sigma = \frac{1}{5}$, $T = 1$, $K = 10 = S_0$.

An example: the characteristic function

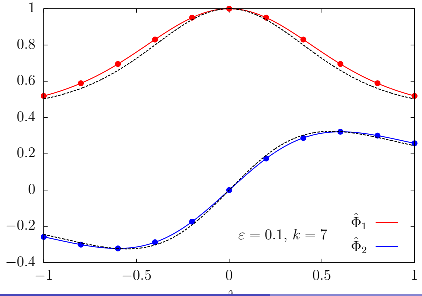
- An SDE model to describe a European call option, where the asset follows

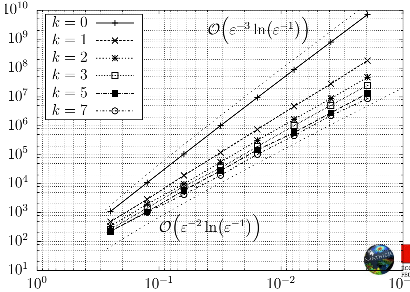
$$dS = rS\,dt + \sigma S\,dW\ , \quad S(0) = S_0\ ,$$

- Quantity of interest is the discounted "payoff": $Q := e^{-rT}\max\big(S(T)-K, 0\big)$
- Approximate **characteristic function** of Q:

$$\Phi(\theta) = \mathbb{E}\big(\cos(\theta Q)\big) + \mathrm{i}\,\mathbb{E}\big(\sin(\theta Q)\big) \equiv \Phi_1(\theta) + \mathrm{i}\,\Phi_2(\theta)\ ,$$

- Milstein scheme with $h_\ell = 2^{-\ell}T$; $\Theta = [-1, 1]$, $r = \frac{1}{20}$, $\sigma = \frac{1}{5}$, $T = 1$, $K = 10 = S_0$.





NASA Common Research Model

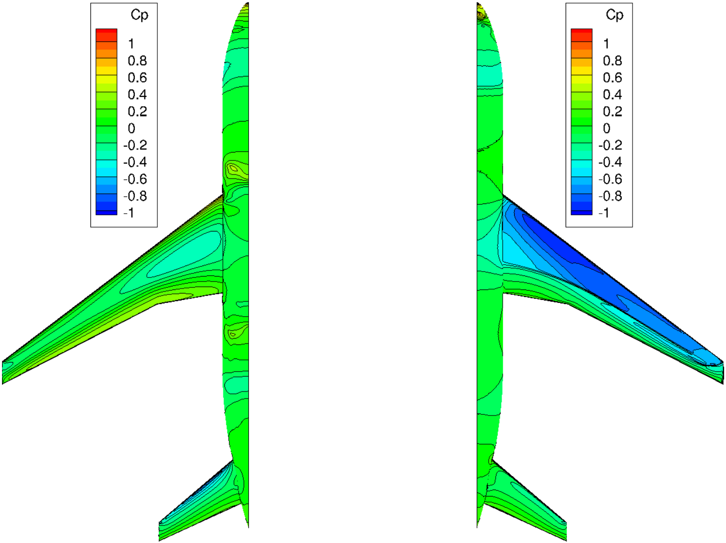
NASA CRM: aircraft configuration equipped with a contemporary supercritical transonic wing and a fuselage that is representative of a wide-body commercial transport aircraft.

Q.ty	**Reference**	**Uncertainty**
M_∞	0.85	$\mathcal{B}(2, 2, 0.05, M_\infty - 0.025)$
Re_c	$5 \cdot 10^6$	–
T_{ref}	310.928 $[K]$	$\mathcal{B}(2, 2, 30, T_{ref} - 15)$
C_L	0.3, 0.4, 0.5, 0.55	–

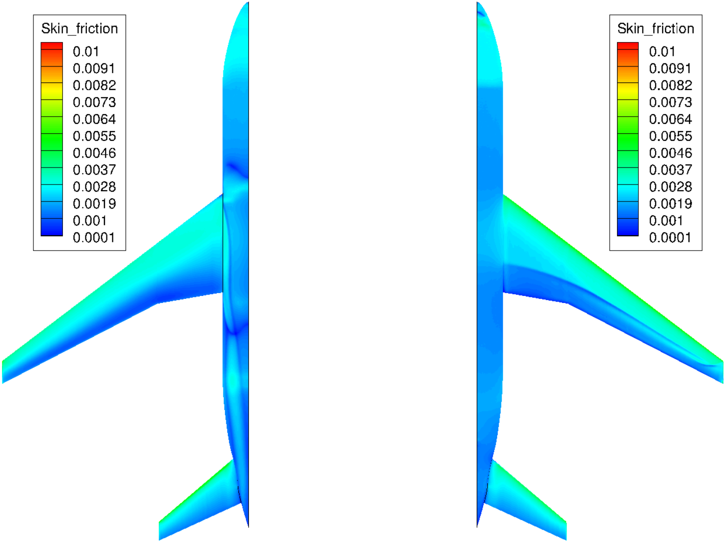
LEVEL	**Cells**	**y+**	*CTime on* 280 *CPUs*
$L0$	$2.3 \cdot 10^6$	$1 - 2$	400 $[s]$ (0.11 $[h]$)
$L1$	$5.0 \cdot 10^6$	$1 - 2$	825 $[s]$ (0.23 $[h]$)
$L2$	$9.8 \cdot 10^6$	$1 - 2$	1250 $[s]$ (0.35 $[h]$)
$L3$	$21.3 \cdot 10^6$	$1 - 2$	3200 $[s]$ (0.89 $[h]$)

Spalart-Allmaras turbulence model, hybrid unstructured grids.

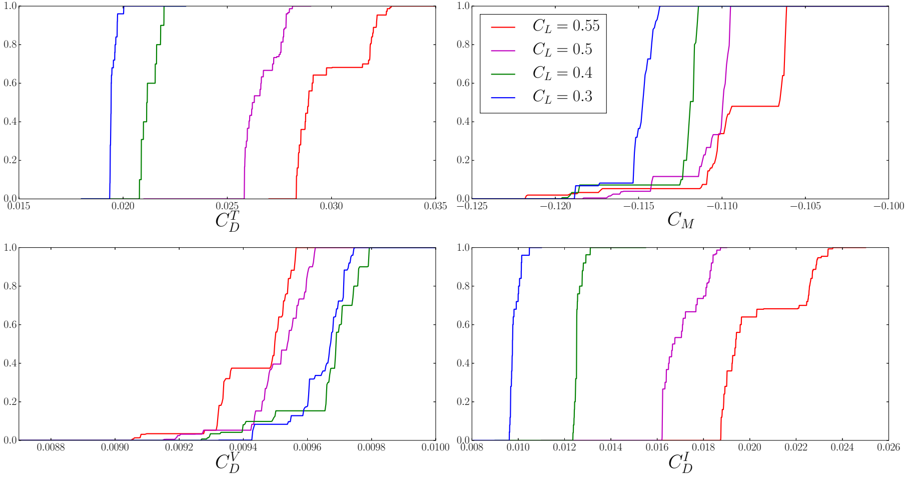
NASA Common Research Model



NASA Common Research Model



NASA Common Research Model



Outline

Risk averse optimization

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

$\mathcal{R}$: risk measure

Examples

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

Risk averse optimization

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

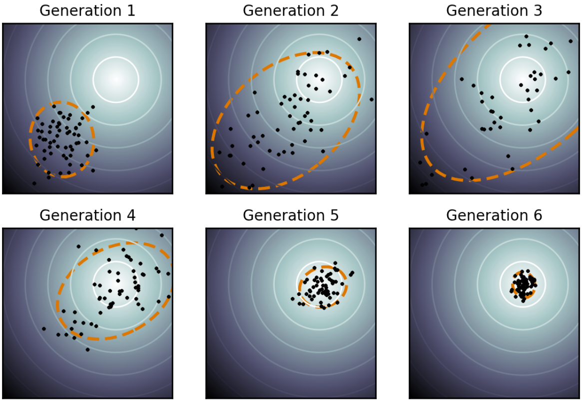
$\mathcal{R}$: risk measure

Examples

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

Combining MLMC with CMA-ES

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



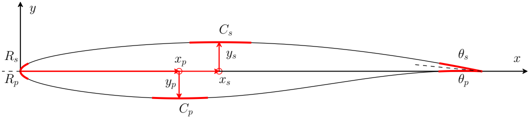
For each individual at each generation, risk measure computed by MLMC.

Airfoil optimization under operating uncertainties

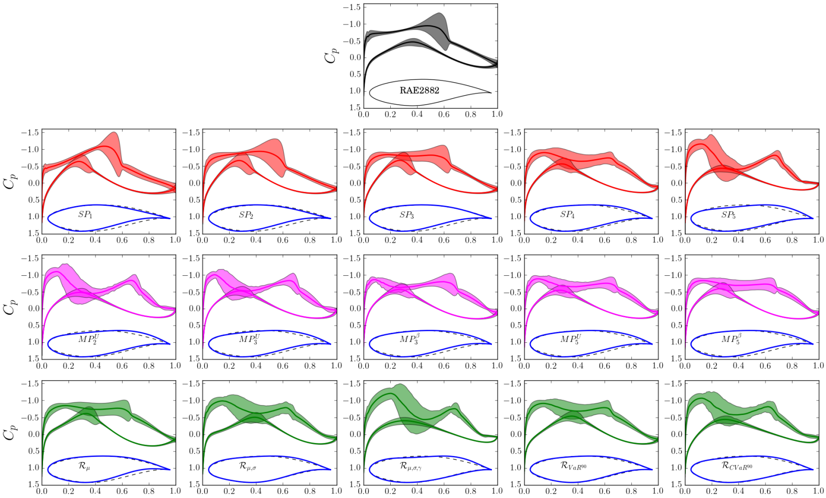
$$\begin{cases} \min_{x \in X} & \mathcal{R}\left[C_D(x)\right] \\ s.t & C_L(x) = C_L^*, \quad \text{thickness constraint} \end{cases}$$

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

	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	–



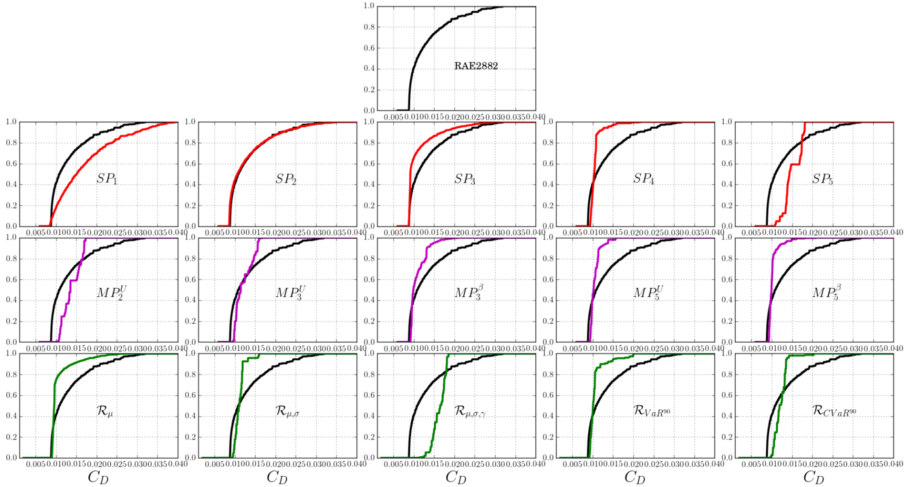
Qualitative comparison



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

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

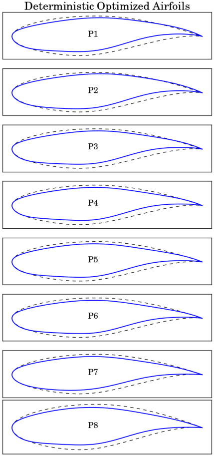
Deterministic versus Robust optimization

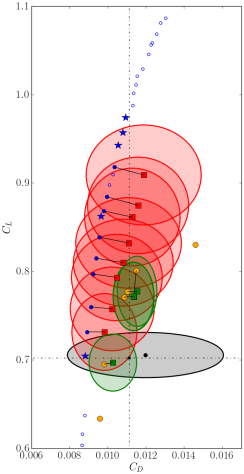


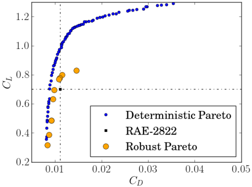
Multi-objective optimization under operating uncertainties

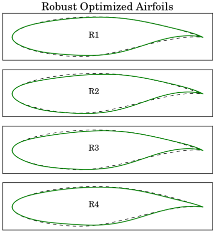
$$\underset{x\in X}{\text{P-min}} \;\; \{\mu_{C_D}(x)+\sigma_{C_D}(x), \;\; -\mu_{C_L}(x)+\sigma_{C_L}(x)\} \qquad \text{(Pareto front)}$$

Uncertainties in Mach number and Angle of Attack.









Outline

Conclusions and outlook

- Multilevel Monte Carlo is a very powerful technique that can dramatically reduce the computational cost of a UQ analysis compared to plain MC.
- The tuning of MLMC requires adaptive algorithms and reliable error and variances estimators.
- We have presented a way to compute higher order moments as well as cdf, quantiles, CVaR with MLMC and properly tune the method.
- The methodology has been successfully applied to forward UQ propagation and robust optimization under uncertainty in compressible aerodynamics.

Thank you for your attention!

References

M. Pisaroni.
Multi Level Monte Carlo Methods for Uncertainty Quantification and Robust Design Optimization in Aerodynamics, PhD Thesis n. 8082, EPFL, 2017

M. Pisaroni, S. Krumscheid, F. Nobile.
Quantifying uncertain system outputs via the multilevel Monte Carlo method - Part I: Central moment estimation, MATHICSE Technical report no. 23.2017.

M. Pisaroni, S. Krumscheid, F. Nobile.
Quantifying uncertain system outputs via the multilevel Monte Carlo method Part 2: distribution and robustness measures, in preparation.

S. Krumscheid, F. Nobile.
Multilevel Monte Carlo approximation of functions, MATHICSE Technical report no. 12.2017.

M. Pisaroni, F. Nobile, P. Leyland.
A Multilevel Monte Carlo Evolutionary Algorithm for Robust Aerodynamic Shape Design. 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Denver, Colorado, USA, 2017.

M. Pisaroni, F. Nobile, P. Leyland.
Continuation Multi-Level Monte-Carlo method for Uncertainty Quantifiction in Turbulent Compressible Aerodynamics Problems modeled by RANS, MATHICSE Technical report no. 10.2017.

M. Pisaroni, F. Nobile, P. Leyland.
A Continuation Multi Level Monte Carlo (C-MLMC) method for uncertainty quantification in compressible inviscid aerodynamics, CMAME, vol. 326, p. 20-50, 2017.

A.-L. Haji-Ali, F. Nobile, R. Tempone.
Multi-index Monte Carlo: when sparsity meets sampling, in Numer. Math., vol. 132(4), p. 767-806, 2016.

N. Collier, A.-L. Haji-Ali, F. Nobile, E. von Schwerin, R. Tempone.
A continuation multilevel Monte Carlo algorithm, BIT Numerical Mathematics, vol. 55(2), p. 399-432, 2015.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE