Multi-level and Multi-index Monte Carlo methods for Uncertainty Quantification

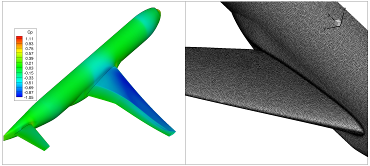
Fabio Nobile

CSQI - Institute of Mathematics, EPFL, Switzerland

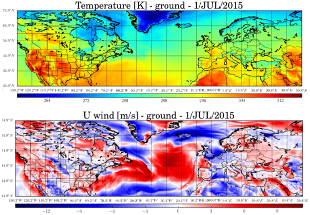
SIAM Conference on Uncertainty Quantification
April 16-19, 2018
Garden Grove, California, USA

Outline

UQ in aerodynamic design



Compute aerodynamic coeffs. (lift, drag, C_p) and optimize airfoil shape in presence of uncertainties

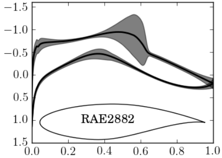


Operational uncertainties

(Mach, angle of attach)

Geometrical uncertainties

(manufacturing, deflection, icing, ...)



UNCERTAINTY MANAGEMENT FOR ROBUST INDUSTRIAL DESIGN IN AERONAUTICS
UMRIDA

MATHICSE

EPFL
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

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

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

In aerospace dynamics problems the response function $y \mapsto Q_h(u(y))$ is often non smooth, with y possibly high dimensional. Monte Carlo type sampling techniques preferred.

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

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

In aerospace dynamics problems the response function $y \mapsto Q_h(u(y))$ is often non smooth, with y possibly high dimensional. Monte Carlo type sampling techniques preferred.

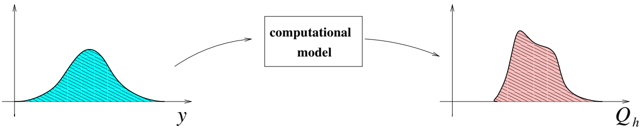
Forward Uncertainty propagation

- 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

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



In aerospace dynamics problems the response function $y \mapsto Q_h(u(y))$ is often non smooth, with y possibly high dimensional. Monte Carlo type sampling techniques preferred.

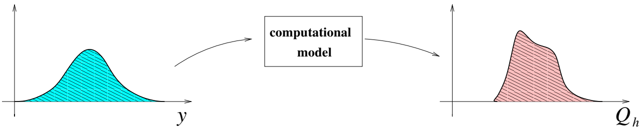
Forward Uncertainty propagation

- 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

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



In aerospace dynamics problems the response function $y \mapsto Q_h(u(y))$ is often non smooth, with y possibly high dimensional. Monte Carlo type sampling techniques preferred.

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

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

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 $Q_h^{(i)}$: $C_h = \mathcal{O}(h^{-\gamma})$

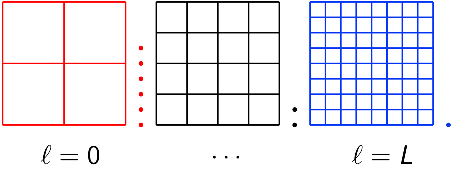
Then $\quad \mathrm{MSE} = \mathcal{O}(tol^2) \quad \Longrightarrow \quad h = \mathcal{O}(tol^{\frac{1}{\alpha}}), \quad M = \mathcal{O}(tol^{-2})$

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

Outline

Multilevel Monte Carlo (MLMC) method

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



- Sequence of refined discretizations

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

- Sequence of sample sizes

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

Telescopic sum (denoting $Q_\ell = Q_{h_\ell}$)

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

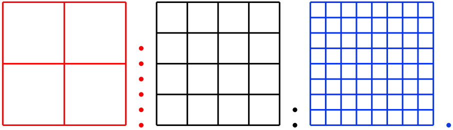
MLMC estimator: estimate each term independently with different sample sizes

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

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

Multilevel Monte Carlo (MLMC) method

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



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

- Sequence of refined discretizations

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

- Sequence of sample sizes

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

Telescopic sum (denoting $Q_\ell = Q_{h_\ell}$)

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

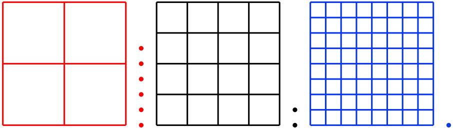
MLMC estimator: estimate each term independently with different sample sizes

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

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

Multilevel Monte Carlo (MLMC) method

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



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

- Sequence of refined discretizations

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

- Sequence of sample sizes

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

Telescopic sum (denoting $Q_\ell = Q_{h_\ell}$)

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

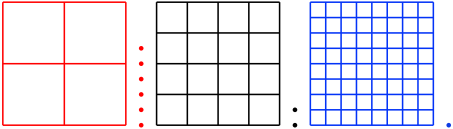
MLMC estimator: estimate each term independently with different sample sizes

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

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

Multilevel Monte Carlo (MLMC) method

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



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

- Sequence of refined discretizations

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

- Sequence of sample sizes

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

Telescopic sum (denoting $Q_\ell = Q_{h_\ell}$)

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

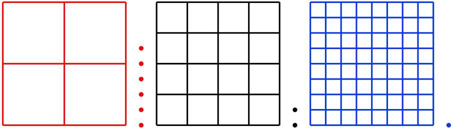
MLMC estimator: estimate each term independently with different sample sizes

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

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

Multilevel Monte Carlo (MLMC) method

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



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

- Sequence of refined discretizations

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

- Sequence of sample sizes

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

Telescopic sum (denoting $Q_\ell = Q_{h_\ell}$)

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

MLMC estimator: estimate each term independently with different sample sizes

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

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

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

- $|\mathbb{E}[Q - Q_\ell]| = \mathcal{O}(h_\ell^\alpha)$,
- $V_\ell = \mathbb{V}\mathrm{ar}[Q_\ell - Q_{\ell-1}] = \mathcal{O}(h_\ell^\beta)$, $\quad (\beta = 2\alpha$ 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

- $V_\ell = \mathbb{V}\text{ar}[Q_\ell - Q_{\ell-1}]$ (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

- $|\mathbb{E}[Q - Q_\ell]| = \mathcal{O}(h_\ell^\alpha)$,
- $V_\ell = \mathbb{V}\text{ar}[Q_\ell - Q_{\ell-1}] = \mathcal{O}(h_\ell^\beta)$, $\quad$ ($\beta = 2\alpha$ 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 – 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 new samples become available.

(Adaptive MLMC [Giles 2015], Continuation MLMC [Collier-HajiAli-N.-vonSchwerin-Tempone 2015])

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 new samples become available.

(Adaptive MLMC [Giles 2015], Continuation MLMC [Collier-HajiAli-N.-vonSchwerin-Tempone 2015])

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 new samples become available.

(Adaptive MLMC [Giles 2015], Continuation MLMC [Collier-HajiAli-N.-vonSchwerin-Tempone 2015])

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 new samples become available.

(Adaptive MLMC [Giles 2015], Continuation MLMC [Collier-HajiAli-N.-vonSchwerin-Tempone 2015])

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 new samples become available.

(Adaptive MLMC [Giles 2015], Continuation MLMC [Collier-HajiAli-N.-vonSchwerin-Tempone 2015])

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

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

$$M_\ell \to \infty \qquad \hat{V}_\ell \approx V_\ell^{MC} \qquad \text{(sample variance)}$$

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

Continuation Multilevel Monte Carlo

[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

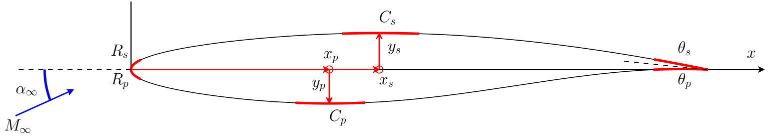
$$M_\ell = 0 \qquad \hat{V}_\ell = V_\ell^{model} \qquad \text{(prior model)}$$

$$M_\ell \to \infty \qquad \hat{V}_\ell \approx V_\ell^{MC} \qquad \text{(sample variance)}$$

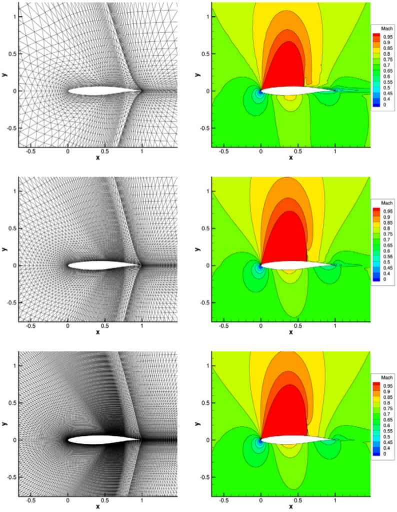
$\hat{V}_\ell$ is then used to determine the sample sizes M_ℓ for the next iteration.

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	-

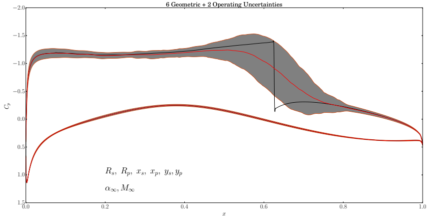


Computation of C_L and pressure coeff. for RAE2822 airfoil



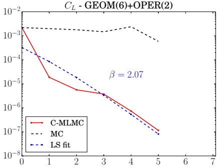
MLMC 5-levels grid hierarchy for the RAE2822 problem.

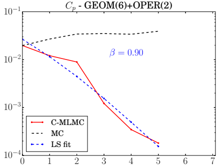
Level	Airfoil nodes	Cells	$\tau(Q_{M_l})[s]$ $(n.cpu)$
$L0$	67	5197	14.4 (18)
$L1$	131	9968	21.4 (22)
$L2$	259	20850	28.8 (28)
$L3$	515	47476	64.0 (36)
$L4$	1027	114857	122.1 (44)
$L5$	2051	283925	314.2 (56)

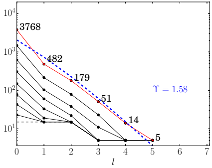


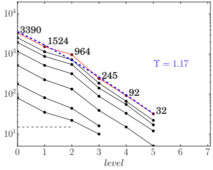
Inviscid model (Euler); SU^2 solver (Stanford) [Pisaroni-N.-Leyland CMAME 2017]

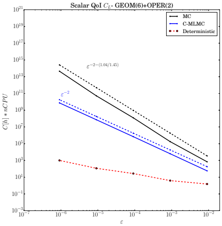
MLMC hierarchies and comparison with MC

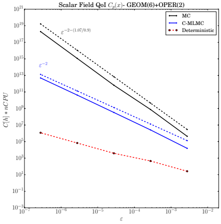


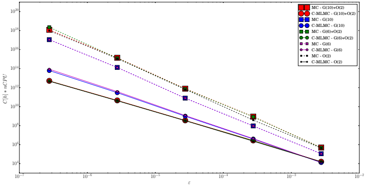




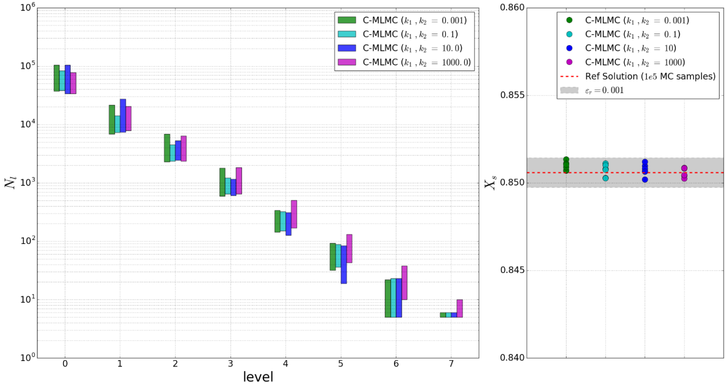








Robustness of C-MLMC estimator



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? use *h*-statistics [Pisaroni-Krumscheid-N. 2017] (alternative approach with biased central moments estimators in [Bierig-Chernov 2015-2016])

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? use *h*-statistics [Pisaroni-Krumscheid-N. 2017] (alternative approach with biased central moments estimators in [Bierig-Chernov 2015-2016])

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? use h-statistics [Pisaroni-Krumscheid-N. 2017] (alternative approach with biased central moments estimators in [Bierig-Chernov 2015-2016])

Given iid sample $\vec{Q}_M = \{Q^{(1)}, \ldots, Q^{(M)}\}$,

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

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? use *h*-statistics [Pisaroni-Krumscheid-N. 2017] (alternative approach with biased central moments estimators in [Bierig-Chernov 2015-2016])

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? use *h*-statistics [Pisaroni-Krumscheid-N. 2017] (alternative approach with biased central moments estimators in [Bierig-Chernov 2015-2016])

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

Practical algorithm: unbiased estimators $\hat{V}_{\ell,p}$ of $V_{\ell,p}$ can be computed in closed form starting from the power terms [Pisaroni-Krumscheid-N. 2017]

$$S_{a,b} = \sum_{i=1}^{M_\ell} (Q^{(i)}_{\ell,M_\ell} + Q^{(i)}_{\ell-1,M_\ell})^a (Q^{(i)}_{\ell,M_\ell} - Q^{(i)}_{\ell-1,M_\ell})^b$$

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.

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

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

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

Beyond expectations: computation of central moments

Practical algorithm: unbiased estimators $\hat{V}_{\ell,p}$ of $V_{\ell,p}$ can be computed in closed form starting from the power terms [Pisaroni-Krumscheid-N. 2017]

$$S_{a,b} = \sum_{i=1}^{M_\ell} (Q^{(i)}_{\ell,M_\ell} + Q^{(i)}_{\ell-1,M_\ell})^a (Q^{(i)}_{\ell,M_\ell} - Q^{(i)}_{\ell-1,M_\ell})^b$$

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.

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

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: 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, $V_\ell = \mathbb{V}\text{ar}[\phi(\theta, Q_\ell) - \phi(\theta, Q_{\ell-1})]$ 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 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.

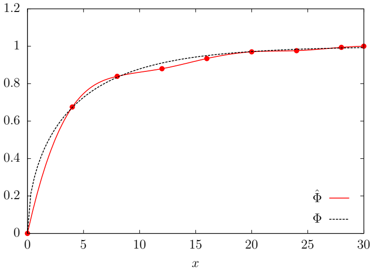
Moreover, from the approximation of Φ_τ and its derivatives we can get for free

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

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

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

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

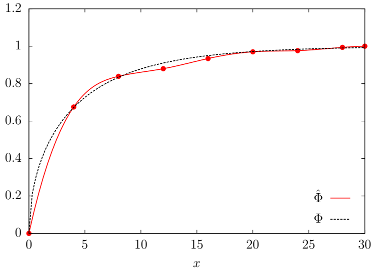
e.g. by spline or polynomial interpolation

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

A practical algorithm to tune the MLMC hierarchy and achieve $\mathrm{MSE}(\hat{\Phi}_L) := \mathbb{E}[\sup_{\theta\in\Theta} |\Phi(\theta) - \hat{\Phi}_L(\theta)|^2] \leq tol$ is proposed in [Pisaroni-Krumscheid-N. in preparation]. Convergence analysis in [Krumscheid-N. 2017].

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

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

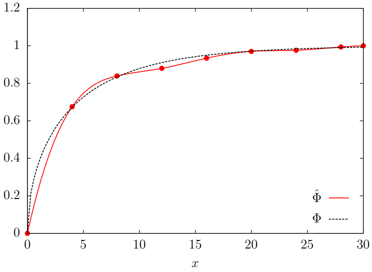
e.g. by spline or polynomial interpolation

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

A practical algorithm to tune the MLMC hierarchy and achieve $\mathrm{MSE}(\hat{\Phi}_L) := \mathbb{E}[\sup_{\theta \in \Theta} |\Phi(\theta) - \hat{\Phi}_L(\theta)|^2] \leq tol$ is proposed in [Pisaroni-Krumscheid-N. in preparation]. Convergence analysis in [Krumscheid-N. 2017].

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

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

e.g. by spline or polynomial interpolation

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

A practical algorithm to tune the MLMC hierarchy and achieve $\mathrm{MSE}(\hat{\Phi}_L) := \mathbb{E}[\sup_{\theta\in\Theta} |\Phi(\theta) - \hat{\Phi}_L(\theta)|^2] \leq tol$ is proposed in [Pisaroni-Krumscheid-N. in preparation]. Convergence analysis in [Krumscheid-N. 2017].

Outline

Risk averse optimization

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

$\mathcal{R}$: risk measure

Examples

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

Risk averse optimization

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

$\mathcal{R}$: risk measure

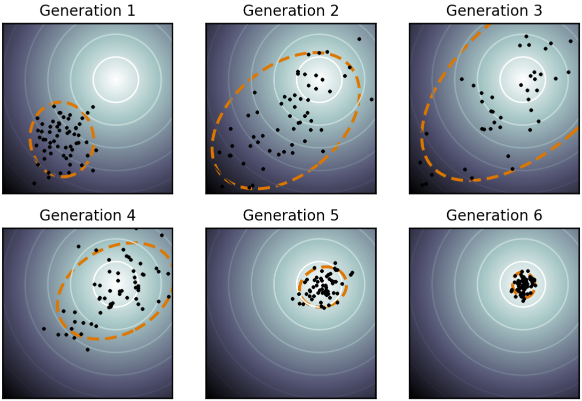
Examples

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

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Combining MLMC with CMA-ES

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



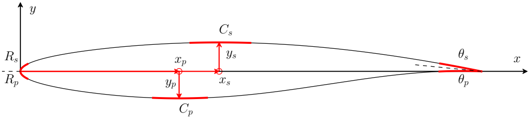
For each individual at each generation, risk measure computed by MLMC.

Airfoil optimization under operating uncertainties

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

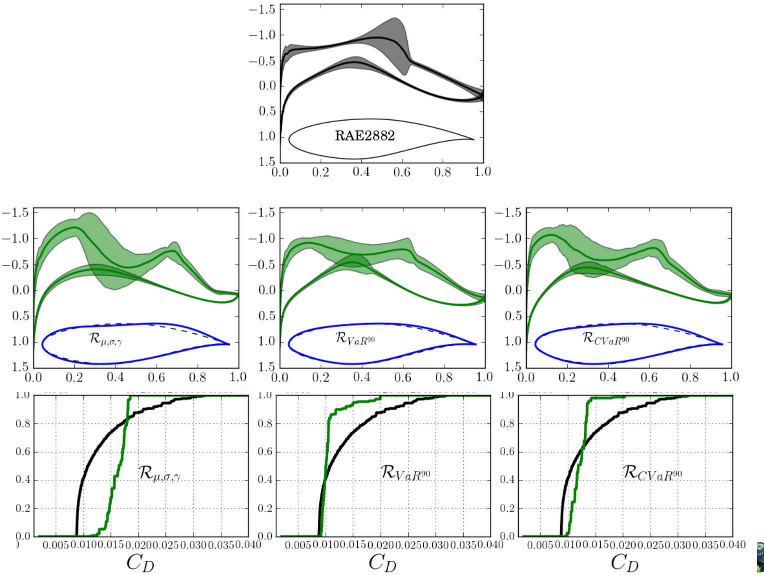
$\mathcal{R}_{\mu,\sigma}\left[C_D(x)\right]$	$\mu_{C_D}(x) + \sigma_{C_D}(x)$
$\mathcal{R}_{\mu,\sigma,\gamma}\left[C_D(x)\right]$	$\mu_{C_D}(x) + \sigma_{C_D}(x) + \mu_{C_D}(x) \cdot \gamma_{C_D}(x)$
$\mathcal{R}_{VaR^{90}}\left[C_D(x)\right]$	$VaR^{90}_{C_D}(x)$
$\mathcal{R}_{CVaR^{90}}\left[C_D(x)\right]$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

	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	–



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

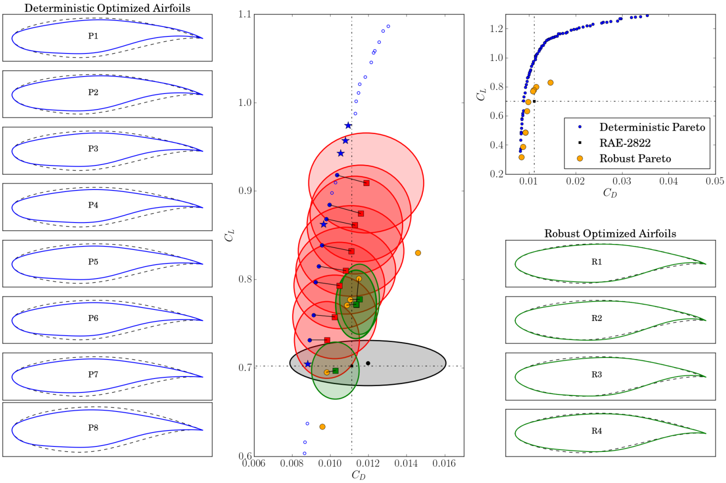
Deterministic versus Robust optimization



Multi-objective optimization under operating uncertainties

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

Uncertainties in Mach number and Angle of Attack.

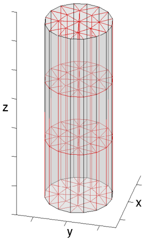


Outline

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



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

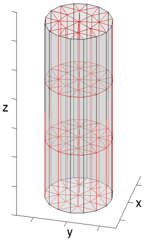
$$\Delta_j Q_{\vec{\ell}} = \begin{cases} Q_{\vec{\ell}} - Q_{\vec{\ell} - \vec{e}_j}, & \text{if } \ell_j > 0 \\ Q_{\vec{\ell}}, & \text{if } \ell = 0 \end{cases}$$

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

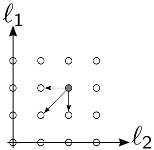
Multi Index Monte Carlo method

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

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

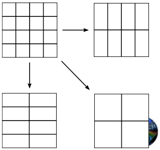


- Introduce sequences of refined discretizations: $h_0^{(i)} > h_0^{(i)} > \ldots > h_{L_i}^{(i)}$
- For $\vec{\ell} = (\ell_1, \dots, \ell_d)$, denote $Q_{\vec{\ell}} = Q(u_{h^{(1)}_{\ell_1}, \dots, h^{(d)}_{\ell_d}})$
- Difference operators



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

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



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

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

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

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

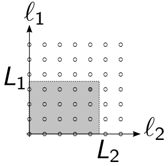
Multi Index Monte Carlo method

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

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



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

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

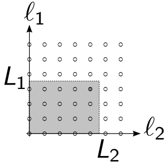
Multi Index Monte Carlo method

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

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

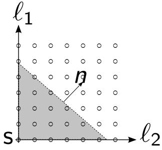
Multi Index idea: compute each expectation independently

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



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

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



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 $\#\{j : \frac{\gamma_j - \beta_j}{\alpha_j} = \max_k \frac{\gamma_k - \beta_k}{\alpha_k} c\}$

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 $\#\{j : \frac{\gamma_j - \beta_j}{\alpha_j} = \max_k \frac{\gamma_k - \beta_k}{\alpha_k} c\}$

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]

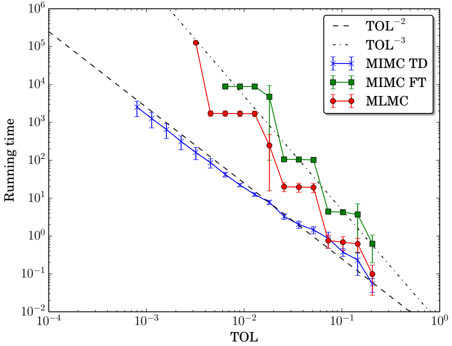
Under the above assumptions, for any $tol > 0$ there exist L and $\{M_{\vec{\ell}}\}_{\vec{\ell} \in \mathcal{I}_L}$ such that $MSE(\mu^{MIMC}{}_{\mathcal{I}_L}) \leq tol^2$ and

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

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

Numerical test

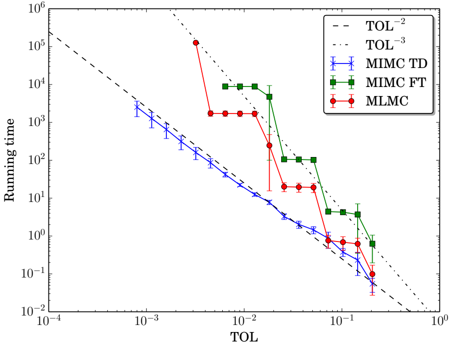
Elliptic equation in 3D with random coefficient and forcing term. Discretization parameters: mesh sizes in the 3 variables (x, y, z) separately.



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].

Outline

Multilevel methods in data assimilation

Lot of recent literature (non-exhaustive list)

- Bayesian inverse problems [Dodwell-Ketelsen-Scheichl-Teckentrup, 2015], [Hoang-Schwab-Stuart, 2013], [Jasra-Jo-Nott-Shoemaker-Tempone, 2017], [Jasra-Kamatani-Law-Zhou, 2018]
- Particle filtering [Jasra-Kamatani-Law-Zhou, 2017]
- Sequential Monte Carlo [Jasra, 2016], [Beskos-Jasra-Law-Tempone-Zhou, 2017], [Beskos-Jasra-Law-Marzouk-Zhou, 2017], [DelMoral-Jasra-Law, 2017], [Latz-Papaioannou-Ullmann, 2018]
- Ensemble Kalman Filter [Hoel-Law-Tempone 2016], [Chernov-Hoel-Law-N.-Tempone 2017]

Filtering problem

- $(\Omega, \mathcal{F}, \mathbb{P})$ complete probability space
- V: separable Hilbert space of "smooth" functions on $D \subset \mathbb{R}^d$ (e.g. $H^s(D), s > 0$)
- $\mathcal{V} \supset V$: separable Hilbert space (weaker than V, e.g. $L^2(D)$)

Dynamics: (Spatio-temporal random process)

$$u^n = \Psi(u^{n-1}), \quad n = 1, 2, \ldots, \qquad \Psi \text{ and / or } u^0 \text{ random}$$

- $u^0 \in L^p(\Omega, V)$, $p \geq 2$;
- $\Psi : L^p(\Omega, V) \mapsto L^p(\Omega, V)$ and $\Psi : L^p(\Omega, \mathcal{V}) \mapsto L^p(\Omega, \mathcal{V})$, Lipschitz continuous

Observations: $y^n = Hu^n + \eta^n, \qquad \eta^n \sim N(0, \Gamma), \quad H : \mathcal{V} \to \mathbb{R}^m$

Goal:

- approximate conditional distribution of $\hat{u}^n = u^n | y^1, \ldots, y^n$ (filtering distr.)
- compute conditional expectations of functionals: $\mathbb{E}[Q(\hat{u}^n)]$, with $Q : \mathcal{V} \mapsto \mathbb{R}$ Lipschitz continuous

Filtering problem

- $(\Omega, \mathcal{F}, \mathbb{P})$ complete probability space
- V: separable Hilbert space of "smooth" functions on $D \subset \mathbb{R}^d$ (e.g. $H^s(D), s > 0$)
- $\mathcal{V} \supset V$: separable Hilbert space (weaker than V, e.g. $L^2(D)$)

Dynamics: (Spatio-temporal random process)

$$u^n = \Psi(u^{n-1}), \quad n = 1, 2, \ldots, \qquad \Psi \text{ and / or } u^0 \text{ random}$$

- $u^0 \in L^p(\Omega, V)$, $p \geq 2$;
- $\Psi : L^p(\Omega, V) \mapsto L^p(\Omega, V)$ and $\Psi : L^p(\Omega, \mathcal{V}) \mapsto L^p(\Omega, \mathcal{V})$, Lipschitz continuous

Observations: $y^n = Hu^n + \eta^n, \qquad \eta^n \sim N(0, \Gamma), \quad H : \mathcal{V} \to \mathbb{R}^m$

Goal:

- approximate conditional distribution of $\hat{u}^n = u^n | y^1, \ldots, y^n$ (filtering distr.)
- compute conditional expectations of functionals: $\mathbb{E}[Q(\hat{u}^n)]$, with $Q : \mathcal{V} \mapsto \mathbb{R}$ Lipschitz continuous

Filtering problem

- $(\Omega, \mathcal{F}, \mathbb{P})$ complete probability space
- V: separable Hilbert space of "smooth" functions on $D \subset \mathbb{R}^d$ (e.g. $H^s(D), s > 0$)
- $\mathcal{V} \supset V$: separable Hilbert space (weaker than V, e.g. $L^2(D)$)

Dynamics: (Spatio-temporal random process)

$$u^n = \Psi(u^{n-1}), \quad n = 1, 2, \dots, \qquad \Psi \text{ and / or } u^0 \text{ random}$$

- $u^0 \in L^p(\Omega, V)$, $p \geq 2$;
- $\Psi : L^p(\Omega, V) \mapsto L^p(\Omega, V)$ and $\Psi : L^p(\Omega, \mathcal{V}) \mapsto L^p(\Omega, \mathcal{V})$, Lipschitz continuous

Observations: $y^n = Hu^n + \eta^n, \qquad \eta^n \sim N(0, \Gamma), \quad H : \mathcal{V} \to \mathbb{R}^m$

Goal:

- approximate conditional distribution of $\hat{u}^n = u^n | y^1, \dots, y^n$ (filtering distr.)
- compute conditional expectations of functionals: $\mathbb{E}[Q(\hat{u}^n)]$, with $Q : \mathcal{V} \mapsto \mathbb{R}$ Lipschitz continuous

Filtering problem

- $(\Omega, \mathcal{F}, \mathbb{P})$ complete probability space
- V: separable Hilbert space of "smooth" functions on $D \subset \mathbb{R}^d$ (e.g. $H^s(D), s > 0$)
- $\mathcal{V} \supset V$: separable Hilbert space (weaker than V, e.g. $L^2(D)$)

Dynamics: (Spatio-temporal random process)

$$u^n = \Psi(u^{n-1}), \quad n = 1, 2, \dots, \qquad \Psi \text{ and / or } u^0 \text{ random}$$

- $u^0 \in L^p(\Omega, V)$, $p \geq 2$;
- $\Psi : L^p(\Omega, V) \mapsto L^p(\Omega, V)$ and $\Psi : L^p(\Omega, \mathcal{V}) \mapsto L^p(\Omega, \mathcal{V})$, Lipschitz continuous

Observations: $y^n = Hu^n + \eta^n, \qquad \eta^n \sim N(0, \Gamma), \quad H : \mathcal{V} \to \mathbb{R}^m$

Goal:

- approximate conditional distribution of $\hat{u}^n = u^n | y^1, \dots, y^n$ (filtering distr.)
- compute conditional expectations of functionals: $\mathbb{E}[Q(\hat{u}^n)]$, with $Q : \mathcal{V} \mapsto \mathbb{R}$ Lipschitz continuous

Mean Field Ensemble Kalman Filter

Computing the full conditional process $\hat{u}^n$ is often out of reach. We consider a surrogate conditional process $\hat{v}^n$ based on ensemble Kalman Filter updates.

Prediction step

$$v^n = \Psi(\hat{v}^{n-1}), \quad n = 1, 2, \ldots, \quad \hat{v}^0 = u^0$$

Compute Covariance operator $C^n \in V \otimes V$ (equiv. $C^n : V' \mapsto V$)

$$C^n = \mathrm{Cov}[v^n] = \mathbb{E}[v^n \otimes v^n] - \mathbb{E}[v^n] \otimes \mathbb{E}[v^n]$$

Compute Kalman gain $K^n : \mathbb{R}^m \to V$

$$K^n = C^n H^* (\Gamma + H C^n H^*)^{-1}$$

Update step (Kalman Filter formula)

$$\hat{v}^n = v^n + K^n(\tilde{y}^n - H v^n)$$

with perturbed measurements $\tilde{y}^n = y^n + \tilde{\eta}^n, \quad \tilde{\eta}^n \overset{\text{iid}}{\sim} N(0, \Gamma)$

In practice:

- Dynamics can not be solved exactly. Introduce sequence of space-time approx. Ψ_ℓ, $\ell = 0, 1, \ldots, L$ on nested finite dim. spaces $\mathcal{V}_\ell \subset \mathcal{V}$
- $\mathrm{Cov}[v^n]$, $\mathbb{E}[Q(\hat{v}^n)]$ can not be computed exactly either. Use multilevel Monte Carlo formula

Mean Field Ensemble Kalman Filter

Computing the full conditional process $\hat{u}^n$ is often out of reach. We consider a surrogate conditional process $\hat{v}^n$ based on ensemble Kalman Filter updates.

Prediction step

$$v^n = \Psi(\hat{v}^{n-1}), \quad n = 1, 2, \ldots, \quad \hat{v}^0 = u^0$$

Compute Covariance operator $C^n \in V \otimes V$ (equiv. $C^n : V' \mapsto V$)

$$C^n = \mathrm{Cov}[v^n] = \mathbb{E}[v^n \otimes v^n] - \mathbb{E}[v^n] \otimes \mathbb{E}[v^n]$$

Compute Kalman gain $K^n : \mathbb{R}^m \to V$

$$K^n = C^n H^* (\Gamma + H C^n H^*)^{-1}$$

Update step (Kalman Filter formula)

$$\hat{v}^n = v^n + K^n(\tilde{y}^n - H v^n)$$

with perturbed measurements $\tilde{y}^n = y^n + \tilde{\eta}^n, \quad \tilde{\eta}^n \overset{\text{iid}}{\sim} N(0, \Gamma)$

In practice:

- Dynamics can not be solved exactly. Introduce sequence of space-time approx. Ψ_ℓ, $\ell = 0, 1, \ldots, L$ on nested finite dim. spaces $\mathcal{V}_\ell \subset \mathcal{V}$
- $\mathrm{Cov}[v^n]$, $\mathbb{E}[Q(\hat{v}^n)]$ can not be computed exactly either. Use multilevel Monte Carlo formula

Mean Field Ensemble Kalman Filter

Computing the full conditional process $\hat{u}^n$ is often out of reach. We consider a surrogate conditional process $\hat{v}^n$ based on ensemble Kalman Filter updates.

Prediction step

$$v^n = \Psi(\hat{v}^{n-1}), \quad n = 1, 2, \dots, \quad \hat{v}^0 = u^0$$

Compute Covariance operator $C^n \in V \otimes V$ (equiv. $C^n : V' \mapsto V$)

$$C^n = \mathrm{Cov}[v^n] = \mathbb{E}[v^n \otimes v^n] - \mathbb{E}[v^n] \otimes \mathbb{E}[v^n]$$

Compute Kalman gain $K^n : \mathbb{R}^m \to V$

$$K^n = C^n H^* (\Gamma + H C^n H^*)^{-1}$$

Update step (Kalman Filter formula)

$$\hat{v}^n = v^n + K^n(\tilde{y}^n - H v^n)$$

with perturbed measurements $\tilde{y}^n = y^n + \tilde{\eta}^n, \quad \tilde{\eta}^n \overset{\text{iid}}{\sim} N(0, \Gamma)$

In practice:

- Dynamics can not be solved exactly. Introduce sequence of space-time approx. Ψ_ℓ, $\ell = 0, 1, \dots, L$ on nested finite dim. spaces $\mathcal{V}_\ell \subset \mathcal{V}$
- $\mathrm{Cov}[v^n]$, $\mathbb{E}[Q(\hat{v}^n)]$ can not be computed exactly either. Use multilevel Monte Carlo formula

Mean Field Ensemble Kalman Filter

Computing the full conditional process $\hat{u}^n$ is often out of reach. We consider a surrogate conditional process $\hat{v}^n$ based on ensemble Kalman Filter updates.

Prediction step

$$v^n = \Psi(\hat{v}^{n-1}), \quad n = 1, 2, \dots, \quad \hat{v}^0 = u^0$$

Compute Covariance operator $C^n \in V \otimes V$ (equiv. $C^n : V' \mapsto V$)

$$C^n = \mathrm{Cov}[v^n] = \mathbb{E}[v^n \otimes v^n] - \mathbb{E}[v^n] \otimes \mathbb{E}[v^n]$$

Compute Kalman gain $K^n : \mathbb{R}^m \to V$

$$K^n = C^n H^* (\Gamma + H C^n H^*)^{-1}$$

Update step (Kalman Filter formula)

$$\hat{v}^n = v^n + K^n(\tilde{y}^n - H v^n)$$

with perturbed measurements $\tilde{y}^n = y^n + \tilde{\eta}^n, \quad \tilde{\eta}^n \overset{\text{iid}}{\sim} N(0, \Gamma)$

In practice:

- Dynamics can not be solved exactly. Introduce sequence of space-time approx. Ψ_ℓ, $\ell = 0, 1, \dots, L$ on nested finite dim. spaces $\mathcal{V}_\ell \subset \mathcal{V}$
- $\mathrm{Cov}[v^n]$, $\mathbb{E}[Q(\hat{v}^n)]$ can not be computed exactly either. Use multilevel Monte Carlo formula

Mean Field Ensemble Kalman Filter

Computing the full conditional process $\hat{u}^n$ is often out of reach. We consider a surrogate conditional process $\hat{v}^n$ based on ensemble Kalman Filter updates.

Prediction step

$$v^n = \Psi(\hat{v}^{n-1}), \quad n = 1, 2, \ldots, \quad \hat{v}^0 = u^0$$

Compute Covariance operator $C^n \in V \otimes V$ (equiv. $C^n : V' \mapsto V$)

$$C^n = \mathrm{Cov}[v^n] = \mathbb{E}[v^n \otimes v^n] - \mathbb{E}[v^n] \otimes \mathbb{E}[v^n]$$

Compute Kalman gain $K^n : \mathbb{R}^m \to V$

$$K^n = C^n H^* (\Gamma + H C^n H^*)^{-1}$$

Update step (Kalman Filter formula)

$$\hat{v}^n = v^n + K^n(\tilde{y}^n - H v^n)$$

with perturbed measurements $\tilde{y}^n = y^n + \tilde{\eta}^n, \quad \tilde{\eta}^n \overset{\text{iid}}{\sim} N(0, \Gamma)$

In practice:

- Dynamics can not be solved exactly. Introduce sequence of space-time approx. Ψ_ℓ, $\ell = 0, 1, \ldots, L$ on nested finite dim. spaces $\mathcal{V}_\ell \subset \mathcal{V}$
- $\mathrm{Cov}[v^n]$, $\mathbb{E}[Q(\hat{v}^n)]$ can not be computed exactly either. Use multilevel Monte Carlo formula

Mean Field Ensemble Kalman Filter

Computing the full conditional process $\hat{u}^n$ is often out of reach. We consider a surrogate conditional process $\hat{v}^n$ based on ensemble Kalman Filter updates.

Prediction step

$$v^n = \Psi(\hat{v}^{n-1}), \quad n = 1, 2, \dots, \quad \hat{v}^0 = u^0$$

Compute Covariance operator $C^n \in V \otimes V$ (equiv. $C^n : V' \mapsto V$)

$$C^n = \mathrm{Cov}[v^n] = \mathbb{E}[v^n \otimes v^n] - \mathbb{E}[v^n] \otimes \mathbb{E}[v^n]$$

Compute Kalman gain $K^n : \mathbb{R}^m \to V$

$$K^n = C^n H^* (\Gamma + H C^n H^*)^{-1}$$

Update step (Kalman Filter formula)

$$\hat{v}^n = v^n + K^n(\tilde{y}^n - H v^n)$$

with perturbed measurements $\tilde{y}^n = y^n + \tilde{\eta}^n, \quad \tilde{\eta}^n \overset{\text{iid}}{\sim} N(0, \Gamma)$

In practice:

- Dynamics can not be solved exactly. Introduce sequence of space-time approx. Ψ_ℓ, $\ell = 0, 1, \dots, L$ on nested finite dim. spaces $\mathcal{V}_\ell \subset \mathcal{V}$
- $\mathrm{Cov}[v^n]$, $\mathbb{E}[Q(\hat{v}^n)]$ can not be computed exactly either. Use multilevel Monte Carlo formula

Multilevel Ensemble Kalman Filter (MLEnKF)

[Hoel-Law-Tempone 2016] (finite dim. case), [Chernov-Hoel-Law-N.-Tempone 2017] (∞ dim. case)

On each discretization level $\ell = 0, 1, \dots, L$ consider M_ℓ coupled particles $(v_\ell^{(i)}, v_{\ell-1}^{(i)}) \in \mathcal{V}_\ell \times \mathcal{V}_{\ell-1}$, $i = 1, \dots, M_\ell$.

Prediction step: $v_\ell^{n,(i)} = \Psi_\ell(\hat{v}_\ell^{n-1,(i)}, \omega^{(i,\ell)})$, $\quad \hat{v}_\ell^{0,(i)} = \Pi_\ell u^0$

$v_{\ell-1}^{n,(i)} = \Psi_{\ell-1}(\hat{v}_{\ell-1}^{n-1,(i)}, \omega^{(i,\ell)})$, $\quad \hat{v}_{\ell-1}^{0,(i)} = \Pi_{\ell-1} u^0$

Compute covariance by ML formula (with $\mathring{v}_\ell^{n,(i)} = v_\ell^{n,(i)} - \frac{1}{M_\ell}\sum_{j=1}^{M_\ell} v_\ell^{n,(j)}$)

$$C_{ML}^n = \sum_{\ell=0}^{L} \frac{1}{M_\ell - 1} \sum_{i=1}^{M_\ell} \left[\mathring{v}_\ell^{n,(i)} \otimes \mathring{v}_\ell^{n,(i)} - \mathring{v}_{\ell-1}^{n,(i)} \otimes \mathring{v}_{\ell-1}^{n,(i)} \right] \in \mathcal{V}_L \otimes \mathcal{V}_L$$

Compute Kalman gain: $K_{ML}^n = C_{ML}^n H^* (\Gamma + (H C_{ML}^n H^*)_+)^{-1} : \mathbb{R}^m \to \mathcal{V}_L$

Update particles positions

$$\hat{v}_s^{n,(i)} = v_s^{n,(i)} + \Pi_s K_{ML}^n (\tilde{y}^{n,(i)} - H v_s^{n,(i)}), \qquad s = \ell, \ell - 1$$

with perturbed measurements $\tilde{y}^{n,(i)} = y^n + \tilde{\eta}^{n,(i)}$, $\quad \tilde{\eta}^{n,(i)} \overset{\text{iid}}{\sim} N(0, \Gamma)$

Compute cond. expectation: $\hat{\mu}_{ML}^n[Q] = \sum_\ell \frac{1}{M_\ell} \sum_i Q(\hat{v}_\ell^{n,(i)}) - Q(\hat{v}_{\ell-1}^{n,(i)})$

Multilevel Ensemble Kalman Filter (MLEnKF)

[Hoel-Law-Tempone 2016] (finite dim. case), [Chernov-Hoel-Law-N.-Tempone 2017] (∞ dim. case)

On each discretization level $\ell = 0, 1, \ldots, L$ consider M_ℓ coupled particles $(v_\ell^{(i)}, v_{\ell-1}^{(i)}) \in \mathcal{V}_\ell \times \mathcal{V}_{\ell-1}$, $i = 1, \ldots, M_\ell$.

Prediction step:

$$v_\ell^{n,(i)} = \Psi_\ell(\hat{v}_\ell^{n-1,(i)}, \omega^{(i,\ell)}), \qquad \hat{v}_\ell^{0,(i)} = \Pi_\ell u^0$$
$$v_{\ell-1}^{n,(i)} = \Psi_{\ell-1}(\hat{v}_{\ell-1}^{n-1,(i)}, \omega^{(i,\ell)}), \qquad \hat{v}_{\ell-1}^{0,(i)} = \Pi_{\ell-1} u^0$$

Compute covariance by ML formula (with $\mathring{v}_\ell^{n,(i)} = v_\ell^{n,(i)} - \frac{1}{M_\ell}\sum_{j=1}^{M_\ell} v_\ell^{n,(j)}$)

$$C_{ML}^n = \sum_{\ell=0}^{L} \frac{1}{M_\ell - 1} \sum_{i=1}^{M_\ell} \left[\mathring{v}_\ell^{n,(i)} \otimes \mathring{v}_\ell^{n,(i)} - \mathring{v}_{\ell-1}^{n,(i)} \otimes \mathring{v}_{\ell-1}^{n,(i)} \right] \in \mathcal{V}_L \otimes \mathcal{V}_L$$

Compute Kalman gain: $K_{ML}^n = C_{ML}^n H^* (\Gamma + (H C_{ML}^n H^*)_+)^{-1} : \mathbb{R}^m \to \mathcal{V}_L$

Update particles positions

$$\hat{v}_s^{n,(i)} = v_s^{n,(i)} + \Pi_s K_{ML}^n (\tilde{y}^{n,(i)} - H v_s^{n,(i)}), \qquad s = \ell, \ell - 1$$

with perturbed measurements $\tilde{y}^{n,(i)} = y^n + \tilde{\eta}^{n,(i)}$, $\tilde{\eta}^{n,(i)} \overset{\text{iid}}{\sim} N(0, \Gamma)$

Compute cond. expectation: $\hat{\mu}_{ML}^n[Q] = \sum_\ell \frac{1}{M_\ell} \sum_i Q(\hat{v}_\ell^{n,(i)}) - Q(\hat{v}_{\ell-1}^{n,(i)})$

Multilevel Ensemble Kalman Filter (MLEnKF)

[Hoel-Law-Tempone 2016] (finite dim. case), [Chernov-Hoel-Law-N.-Tempone 2017] (∞ dim. case)

On each discretization level $\ell = 0, 1, \dots, L$ consider M_ℓ coupled particles $(v_\ell^{(i)}, v_{\ell-1}^{(i)}) \in \mathcal{V}_\ell \times \mathcal{V}_{\ell-1}$, $i = 1, \dots, M_\ell$.

Prediction step:

$$v_\ell^{n,(i)} = \Psi_\ell(\hat{v}_\ell^{n-1,(i)}, \omega^{(i,\ell)}), \qquad \hat{v}_\ell^{0,(i)} = \Pi_\ell u^0$$

$$v_{\ell-1}^{n,(i)} = \Psi_{\ell-1}(\hat{v}_{\ell-1}^{n-1,(i)}, \omega^{(i,\ell)}), \qquad \hat{v}_{\ell-1}^{0,(i)} = \Pi_{\ell-1} u^0$$

Compute covariance by ML formula (with $\mathring{v}_\ell^{n,(i)} = v_\ell^{n,(i)} - \frac{1}{M_\ell}\sum_{j=1}^{M_\ell} v_\ell^{n,(j)}$)

$$C_{ML}^n = \sum_{\ell=0}^{L} \frac{1}{M_\ell - 1} \sum_{i=1}^{M_\ell} \left[\mathring{v}_\ell^{n,(i)} \otimes \mathring{v}_\ell^{n,(i)} - \mathring{v}_{\ell-1}^{n,(i)} \otimes \mathring{v}_{\ell-1}^{n,(i)} \right] \in \mathcal{V}_L \otimes \mathcal{V}_L$$

Compute Kalman gain: $K_{ML}^n = C_{ML}^n H^* (\Gamma + (H C_{ML}^n H^*)_+)^{-1} : \mathbb{R}^m \to \mathcal{V}_L$

Update particles positions

$$\hat{v}_s^{n,(i)} = v_s^{n,(i)} + \Pi_s K_{ML}^n (\tilde{y}^{n,(i)} - H v_s^{n,(i)}), \qquad s = \ell, \ell-1$$

with perturbed measurements $\tilde{y}^{n,(i)} = y^n + \tilde{\eta}^{n,(i)}$, $\tilde{\eta}^{n,(i)} \overset{\text{iid}}{\sim} N(0, \Gamma)$

Compute cond. expectation: $\hat{\mu}_{ML}^n[Q] = \sum_\ell \frac{1}{M_\ell} \sum_i Q(\hat{v}_\ell^{n,(i)}) - Q(\hat{v}_{\ell-1}^{n,(i)})$

Multilevel Ensemble Kalman Filter (MLEnKF)

[Hoel-Law-Tempone 2016] (finite dim. case), [Chernov-Hoel-Law-N.-Tempone 2017] (∞ dim. case)

On each discretization level $\ell = 0, 1, \dots, L$ consider M_ℓ coupled particles $(v_\ell^{(i)}, v_{\ell-1}^{(i)}) \in \mathcal{V}_\ell \times \mathcal{V}_{\ell-1}$, $i = 1, \dots, M_\ell$.

Prediction step:

$$v_\ell^{n,(i)} = \Psi_\ell(\hat{v}_\ell^{n-1,(i)}, \omega^{(i,\ell)}), \qquad \hat{v}_\ell^{0,(i)} = \Pi_\ell u^0$$
$$v_{\ell-1}^{n,(i)} = \Psi_{\ell-1}(\hat{v}_{\ell-1}^{n-1,(i)}, \omega^{(i,\ell)}), \quad \hat{v}_{\ell-1}^{0,(i)} = \Pi_{\ell-1} u^0$$

Compute covariance by ML formula (with $\mathring{v}_\ell^{n,(i)} = v_\ell^{n,(i)} - \frac{1}{M_\ell}\sum_{j=1}^{M_\ell} v_\ell^{n,(j)}$)

$$C_{ML}^n = \sum_{\ell=0}^{L} \frac{1}{M_\ell - 1} \sum_{i=1}^{M_\ell} \left[\mathring{v}_\ell^{n,(i)} \otimes \mathring{v}_\ell^{n,(i)} - \mathring{v}_{\ell-1}^{n,(i)} \otimes \mathring{v}_{\ell-1}^{n,(i)} \right] \in \mathcal{V}_L \otimes \mathcal{V}_L$$

Compute Kalman gain: $K_{ML}^n = C_{ML}^n H^* (\Gamma + (H C_{ML}^n H^*)_+)^{-1} : \mathbb{R}^m \to \mathcal{V}_L$

Update particles positions

$$\hat{v}_s^{n,(i)} = v_s^{n,(i)} + \Pi_s K_{ML}^n (\tilde{y}^{n,(i)} - H v_s^{n,(i)}), \qquad s = \ell, \ell-1$$

with perturbed measurements $\tilde{y}^{n,(i)} = y^n + \tilde{\eta}^{n,(i)}$, $\tilde{\eta}^{n,(i)} \overset{\text{iid}}{\sim} N(0, \Gamma)$

Compute cond. expectation: $\hat{\mu}_{ML}^n[Q] = \sum_\ell \frac{1}{M_\ell} \sum_i Q(\hat{v}_\ell^{n,(i)}) - Q(\hat{v}_{\ell-1}^{n,(i)})$

Multilevel Ensemble Kalman Filter (MLEnKF)

[Hoel-Law-Tempone 2016] (finite dim. case), [Chernov-Hoel-Law-N.-Tempone 2017] (∞ dim. case)

On each discretization level $\ell = 0, 1, \dots, L$ consider M_ℓ coupled particles $(v_\ell^{(i)}, v_{\ell-1}^{(i)}) \in \mathcal{V}_\ell \times \mathcal{V}_{\ell-1}$, $i = 1, \dots, M_\ell$.

Prediction step:
$$v_\ell^{n,(i)} = \Psi_\ell(\hat{v}_\ell^{n-1,(i)}, \omega^{(i,\ell)}), \qquad \hat{v}_\ell^{0,(i)} = \Pi_\ell u^0$$
$$v_{\ell-1}^{n,(i)} = \Psi_{\ell-1}(\hat{v}_{\ell-1}^{n-1,(i)}, \omega^{(i,\ell)}), \qquad \hat{v}_{\ell-1}^{0,(i)} = \Pi_{\ell-1} u^0$$

Compute covariance by ML formula (with $\mathring{v}_\ell^{n,(i)} = v_\ell^{n,(i)} - \frac{1}{M_\ell}\sum_{j=1}^{M_\ell} v_\ell^{n,(j)}$)

$$C_{ML}^n = \sum_{\ell=0}^{L} \frac{1}{M_\ell - 1} \sum_{i=1}^{M_\ell} \left[\mathring{v}_\ell^{n,(i)} \otimes \mathring{v}_\ell^{n,(i)} - \mathring{v}_{\ell-1}^{n,(i)} \otimes \mathring{v}_{\ell-1}^{n,(i)} \right] \in \mathcal{V}_L \otimes \mathcal{V}_L$$

Compute Kalman gain: $K_{ML}^n = C_{ML}^n H^* (\Gamma + (H C_{ML}^n H^*)_+)^{-1} : \mathbb{R}^m \to \mathcal{V}_L$

Update particles positions

$$\hat{v}_s^{n,(i)} = v_s^{n,(i)} + \Pi_s K_{ML}^n (\tilde{y}^{n,(i)} - H v_s^{n,(i)}), \qquad s = \ell, \ell-1$$

with perturbed measurements $\tilde{y}^{n,(i)} = y^n + \tilde{\eta}^{n,(i)}$, $\tilde{\eta}^{n,(i)} \overset{\text{iid}}{\sim} N(0, \Gamma)$

Compute cond. expectation: $\hat{\mu}_{ML}^n[Q] = \sum_\ell \frac{1}{M_\ell} \sum_i Q(\hat{v}_\ell^{n,(i)}) - Q(\hat{v}_{\ell-1}^{n,(i)})$

Multilevel Ensemble Kalman Filter (MLEnKF)

[Hoel-Law-Tempone 2016] (finite dim. case), [Chernov-Hoel-Law-N.-Tempone 2017] (∞ dim. case)

On each discretization level $\ell = 0, 1, \dots, L$ consider M_ℓ coupled particles $(v_\ell^{(i)}, v_{\ell-1}^{(i)}) \in \mathcal{V}_\ell \times \mathcal{V}_{\ell-1}$, $i = 1, \dots, M_\ell$.

Prediction step:

$$v_\ell^{n,(i)} = \Psi_\ell(\hat{v}_\ell^{n-1,(i)}, \omega^{(i,\ell)}), \qquad \hat{v}_\ell^{0,(i)} = \Pi_\ell u^0$$

$$v_{\ell-1}^{n,(i)} = \Psi_{\ell-1}(\hat{v}_{\ell-1}^{n-1,(i)}, \omega^{(i,\ell)}), \quad \hat{v}_{\ell-1}^{0,(i)} = \Pi_{\ell-1} u^0$$

Compute covariance by ML formula (with $\mathring{v}_\ell^{n,(i)} = v_\ell^{n,(i)} - \frac{1}{M_\ell}\sum_{j=1}^{M_\ell} v_\ell^{n,(j)}$)

$$C_{ML}^n = \sum_{\ell=0}^{L} \frac{1}{M_\ell - 1} \sum_{i=1}^{M_\ell} \left[\mathring{v}_\ell^{n,(i)} \otimes \mathring{v}_\ell^{n,(i)} - \mathring{v}_{\ell-1}^{n,(i)} \otimes \mathring{v}_{\ell-1}^{n,(i)} \right] \in \mathcal{V}_L \otimes \mathcal{V}_L$$

Compute Kalman gain: $K_{ML}^n = C_{ML}^n H^*(\Gamma + (HC_{ML}^n H^*)_+)^{-1} : \mathbb{R}^m \to \mathcal{V}_L$

Update particles positions

$$\hat{v}_s^{n,(i)} = v_s^{n,(i)} + \Pi_s K_{ML}^n(\tilde{y}^{n,(i)} - Hv_s^{n,(i)}), \qquad s = \ell, \ell-1$$

with perturbed measurements $\tilde{y}^{n,(i)} = y^n + \tilde{\eta}^{n,(i)}$, $\tilde{\eta}^{n,(i)} \overset{\text{iid}}{\sim} N(0,\Gamma)$

Compute cond. expectation: $\hat{\mu}_{ML}^n[Q] = \sum_\ell \frac{1}{M_\ell}\sum_i Q(\hat{v}_\ell^{n,(i)}) - Q(\hat{v}_{\ell-1}^{n,(i)})$

Multilevel Ensemble Kalman Filter (MLEnKF)

Complexity analysis [Chernov-Hoel-Law-N.-Tempone 2017] (generalizes [Hoel-Law-Tempone 2016])

Assume:

- $\inf_{u_\ell \in \mathcal{V}_\ell} \|u - u_\ell\|_{\mathcal{V}} \leq C h_\ell^{\beta/2} \|u\|_V$, $\forall u \in V$
- $\|\Psi(u) - \Psi_\ell(u)\|_{L^p(\Omega,\mathcal{V})} = \mathcal{O}(h_\ell^{\beta/2})$, $\forall u \in L^p(\Omega, V)$
- Cost to compute each pair $(v_\ell^{(i)}, v_{\ell-1}^{(i)})$ is $\mathcal{O}(h_\ell^{-\gamma})$
- Ψ_ℓ Lipschitz continuous in $\mathcal{V}_\ell$ uniformly in ℓ.

Then, for any $tol > 0$ there exist L and $\{M_\ell\}_{\ell=0}^L$ such that

$$\|\hat{\mu}_{ML}^n[Q] - \mathbb{E}[Q(\hat{v}^n)]\|_{L^p(\Omega)} = \mathcal{O}(tol|\log tol|^n)$$

and

$$W(\hat{\mu}_{ML}^n[Q]) \lesssim \begin{cases} tol^{-2} & \text{if } \beta > \gamma \\ tol^{-2}|\log tol|^3 & \text{if } \beta = \gamma \\ tol^{-\frac{\gamma}{\beta/2}} & \text{if } \beta < \gamma \end{cases}$$

Remark: for the standard EnKF we can show the cost-to-accuracy bound

$$W(\mu_{EnKF}^n[Q]) \lesssim tol^{-2-\frac{\gamma}{\beta/2}}$$

A numerical example

Linear stochastic heat equation

$$\begin{cases} du = \Delta u + B dW, & (t,x) \in (0,T] \times (0,1) \\ u(0,x) = u_0(x), & x \in (0,1) \\ u(t,0) = u(t,1) = 0, & t \in (0,T] \end{cases}$$

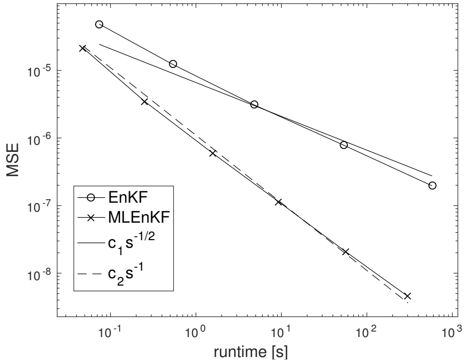
- $\{\phi_j\}_{j=1}^{\infty}$: $L^2(D)$-orthonormal eigenfunctions of $-\Delta$; $\{\lambda_j\}_{j=1}^{\infty}$: eigenvalues
- $B = \sum_j \lambda_j^{-b} \phi_j \otimes \phi_j, \quad b = \frac{1}{2} + \epsilon, \quad \epsilon = 10^{-3}$
- $u_0 = \sum_j j^{-2+\epsilon} \phi_j$
- $y^n = u(t_n, 0.5) + N(0, \frac{1}{4})$
- $Q(u) = \sum_j \hat{u}_j = \sum_j (u, \phi_j)_{L^2}$
- $V = H^{\frac{3+\epsilon}{2}}(0,1), \quad \mathcal{V} = H^{\frac{1+\epsilon}{2}}(0,1)$
- $T = 1/4$, $N = 40$ observation times

A numerical example

Numerical Approximation

- Spectral approx. in space: $\mathcal{V}_\ell = \text{span}\{\phi_1, \dots, \phi_{N_\ell}\}$ with $N_\ell = 2^\ell = h_\ell^{-1}$
- Exponential Euler method in time with $\Delta t_\ell = h_\ell = 2^{-\ell}$
- Assumptions in complexity result verified with $\beta = 2$, $\gamma = 2$

complexity: $W(\hat{\mu}^n_{ML}[Q]) \lesssim tol^{-2} |\log(tol)|^3$



Outline

Conclusions

- Multilevel Monte Carlo can be used effectively to compute expectations, central moments, CDFs, quantiles, superquantiles of output quantities of interest.
- Robust adaptive algorithms are available to tune on the fly the ML hierarchy and control the overall accuracy of the result.
- MLMC methods have been successfully employed in aerodynamic uncertainty quantification and robust airfoil design.
- The Multi-index Monte Carlo construction is a very powerful generalization of the MLMC method and can lead to substantial computational savings whenever mixed type regularities are available for the problem at hand.
- We have proposed a multilevel version of Ensemble Kalman Filter for spatio-temporal processes.

Acknowledgments

- Abdul-Lateef Haji Ali (Oxford)
- Alexey Chernov (Univ. Oldenburg)
- Håkon Hoel (Chalmers Univ.)
- Sebastian Krumscheid (EPFL - CSQI)
- Pénélope Leyland (EPFL - Mechanics)
- Kody Law (Manchester Univ.)
- Michele Pisaroni (EPFL - CSQI)
- Raul Tempone (KAUST)
- Erik von Schwerin (KAUST)

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

Center for Advanced Modeling Science

Thank you for your attention!

References

M. Pisaroni, F. Nobile, P. Leyland.
A continuation-multilevel Monte Carlo evolutionary algorithm for robust aerodynamic shape design, MATHICSE Technical Report no. 06.2018.

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.

A. Chernov, H. Hoel, J.H.K. Law, F. Nobile, R. Tempone.
Multilevel ensemble Kalman filering for spatio-temporal processes, Mathicse Technical Report no. 22.2017.