# A multilevel stochastic gradient algorithm for PDE-constrained optimal control problems under uncertainty

Fabio Nobile

CSQI – Institute of Mathematics, EPFL, Switzerland

**Joint work with**: M. Martin (Criteo, Grenoble), S. Krumscheid (RWTH Aachen), P. Tsilifis (EPFL)

RICAM Special Semestre on Optimization
Workshop 3 "Optimization and Inversion under Uncertainty"
November 11-15, 2019, Linz, Austria

# Outline

1. Problem setting – quadratic optimal control problem
2. Discretization by finite elements + Monte Carlo
3. Deterministic (CG) iterative solvers versus Stochastic Gradient
4. Multilevel stochastic gradient algorithms
5. Conclusions

# Outline

# Problem setting

- $(\Omega, \mathcal{F}, \mathcal{P})$: complete probability space

# Problem setting

- $(\Omega, \mathcal{F}, \mathcal{P})$: complete probability space
- $D \subset \mathbb{R}^d$: physical domain. Throughout $\|\cdot\| = \|\cdot\|_{L^2(D)}$.

# Problem setting

- $(\Omega, \mathcal{F}, \mathcal{P})$: complete probability space
- $D \subset \mathbb{R}^d$: physical domain. Throughout $\|\cdot\| = \|\cdot\|_{L^2(D)}$.
- Forward problem

$$\begin{cases} -\mathrm{div}(a(x,\omega)\nabla y(x,\omega)) = g(x) + u(x), & \text{for a.e. } x \in D,\ \omega \in \Omega \\ y(x,\omega) = 0, & \text{for a.e. } x \in \partial D,\ \omega \in \Omega \end{cases} \qquad (*)$$

with $a(\cdot,\omega)$ a random field s.t. $0 < a_{min} \le a(x,\omega) \le a_{max}$, $\forall (x,\omega) \in D \times \Omega$.

$\Longrightarrow$ random solution $\omega \mapsto y(\cdot,\omega) \in H^1_0(D)$. In particular $y \in L^2_{\mathcal{P}}(\Omega; H^1_0(D))$.

# Problem setting

- $(\Omega, \mathcal{F}, \mathcal{P})$: complete probability space
- $D \subset \mathbb{R}^d$: physical domain. Throughout $\|\cdot\| = \|\cdot\|_{L^2(D)}$.
- Forward problem

$$\begin{cases} -\mathrm{div}(a(x,\omega)\nabla y(x,\omega)) = g(x) + u(x), & \text{for a.e. } x \in D,\ \omega \in \Omega \\ y(x,\omega) = 0, & \text{for a.e. } x \in \partial D,\ \omega \in \Omega \end{cases} \qquad (*)$$

  with $a(\cdot,\omega)$ a random field s.t. $0 < a_{min} \leq a(x,\omega) \leq a_{max}$, $\forall (x,\omega) \in D \times \Omega$.

  $\Longrightarrow$ random solution $\omega \mapsto y(\cdot,\omega) \in H_0^1(D)$. In particular $y \in L^2_{\mathcal{P}}(\Omega; H_0^1(D))$.
- $u \in L^2(D)$: control function

# Problem setting

- $(\Omega, \mathcal{F}, \mathcal{P})$: complete probability space
- $D \subset \mathbb{R}^d$: physical domain. Throughout $\|\cdot\| = \|\cdot\|_{L^2(D)}$.
- Forward problem

$$\begin{cases} -\mathrm{div}(a(x,\omega)\nabla y(x,\omega)) = g(x) + u(x), & \text{for a.e. } x \in D,\ \omega \in \Omega \\ y(x,\omega) = 0, & \text{for a.e. } x \in \partial D,\ \omega \in \Omega \end{cases} \qquad (*)$$

with $a(\cdot,\omega)$ a random field s.t. $0 < a_{min} \le a(x,\omega) \le a_{max}$, $\forall (x,\omega) \in D \times \Omega$.

$\Longrightarrow$ random solution $\omega \mapsto y(\cdot,\omega) \in H_0^1(D)$. In particular $y \in L^2_{\mathcal{P}}(\Omega; H_0^1(D))$.

- $u \in L^2(D)$: control function

## Optimal control problem

$$\min_{\substack{u \in L^2(D) \\ y \in L^2_{\mathcal{P}}(\Omega; H_0^1(D))}} \tilde{J}(u,y) := \frac{1}{2}\mathbb{E}_\omega[\|y(\cdot,\omega) - y_{target}\|^2] + \frac{\beta}{2}\|u\|^2, \qquad \text{subject to } (*)$$

# Reduced functional

- (Stochastic) Affine solution operator: $y_\omega : L^2(D) \to H_0^1(D)$

$$\forall \omega \in \Omega \quad u \mapsto y_\omega(u), \quad \text{solution of } \begin{cases} -\text{div}(a(\cdot,\omega)\nabla y_\omega(u)) = g + u, & \text{in } D \\ y_\omega(u) = 0, & \text{on } \partial D. \end{cases}$$

# Reduced functional

- (Stochastic) Affine solution operator: $y_\omega : L^2(D) \to H_0^1(D)$

$$\forall \omega \in \Omega \quad u \mapsto y_\omega(u), \quad \text{solution of } \begin{cases} -\mathrm{div}(a(\cdot,\omega)\nabla y_\omega(u)) = g + u, & \text{in } D \\ y_\omega(u) = 0, & \text{on } \partial D. \end{cases}$$

- Reduced functional: $\min_{u \in L^2(D)} J(u)$

$$J(u) = \mathbb{E}_\omega[f(u,\omega)], \qquad f(u,\omega) = \frac{1}{2}\|y_\omega(u) - y_{target}\|^2 + \frac{\beta}{2}\|u\|^2$$

# Reduced functional

- (Stochastic) Affine solution operator: $y_\omega : L^2(D) \to H_0^1(D)$

$$\forall \omega \in \Omega \quad u \mapsto y_\omega(u), \quad \text{solution of } \begin{cases} -\mathrm{div}(a(\cdot,\omega)\nabla y_\omega(u)) = g + u, & \text{in } D \\ y_\omega(u) = 0, & \text{on } \partial D. \end{cases}$$

- Reduced functional: $\min_{u \in L^2(D)} J(u)$

$$J(u) = \mathbb{E}_\omega[f(u,\omega)], \qquad f(u,\omega) = \frac{1}{2}\|y_\omega(u) - y_{target}\|^2 + \frac{\beta}{2}\|u\|^2$$

- Adjoint based gradient computation:

$$\nabla_u f(u,\omega) = \beta u + p_\omega(u), \qquad \nabla_u J(u) = \beta u + \mathbb{E}_\omega[p_\omega(u)]$$

where $p_\omega(u)$ solves the adjoint problem $\forall \omega \in \Omega$

$$-\mathrm{div}(a(\cdot,\omega)\nabla p_\omega(u)) = y_\omega(u) - y_{target} \ \text{ in } D, \qquad p_\omega(u) = 0 \ \text{ on } \partial D.$$

# Reduced functional

- (Stochastic) Affine solution operator: $y_\omega : L^2(D) \to H^1_0(D)$

$$\forall \omega \in \Omega \quad u \mapsto y_\omega(u), \quad \text{solution of } \begin{cases} -\text{div}(a(\cdot,\omega)\nabla y_\omega(u)) = g + u, & \text{in } D \\ y_\omega(u) = 0, & \text{on } \partial D. \end{cases}$$

- Reduced functional: $\min_{u \in L^2(D)} J(u)$

$$J(u) = \mathbb{E}_\omega[f(u,\omega)], \qquad f(u,\omega) = \frac{1}{2}\|y_\omega(u) - y_{target}\|^2 + \frac{\beta}{2}\|u\|^2$$

- Adjoint based gradient computation:

$$\nabla_u f(u,\omega) = \beta u + p_\omega(u), \qquad \nabla_u J(u) = \beta u + \mathbb{E}_\omega[p_\omega(u)]$$

where $p_\omega(u)$ solves the adjoint problem $\forall \omega \in \Omega$

$$-\text{div}(a(\cdot,\omega)\nabla p_\omega(u)) = y_\omega(u) - y_{target} \text{ in } D, \qquad p_\omega(u) = 0 \text{ on } \partial D.$$

- Lipschitz and strong convexity properties of $\nabla_u f$: $\forall u_1, u_1 \in L^2(D)$, $\omega \in \Omega$

$$\|\nabla_u f(u_1,\omega) - \nabla_u f(u_2,\omega)\| \le L\|u_1 - u_2\|, \qquad L = \beta + \frac{C_P^4}{a_{min}^2}$$

$$\langle \nabla_u f(u_1,\omega) - \nabla_u f(u_2,\omega), u_1 - u_2\rangle_{L^2(D)} \ge \frac{\ell}{2}\|u_1 - u_2\|^2, \quad \ell = 2\beta$$

# Outline

# Finite dimensional approximation

- Finite Element approximation of the PDE: $\forall u \in L^2(D)$, $\omega \in \Omega$

$$u \mapsto y_\omega^h(u) \quad \text{solves} \quad \int_D a(\cdot,\omega)\nabla y_\omega^h(u)\cdot\nabla v_h = \int_D (g+u)v_h, \quad \forall v_h \in Y_h^r$$

$Y_h^r$: space of continuous $\mathbb{P}_r$ finite element functions vanishing on the boundary

# Finite dimensional approximation

- Finite Element approximation of the PDE: $\forall u \in L^2(D),\ \omega \in \Omega$

$$u \mapsto y_\omega^h(u) \quad \text{solves} \quad \int_D a(\cdot,\omega)\nabla y_\omega^h(u)\cdot\nabla v_h = \int_D (g+u)v_h, \quad \forall v_h \in Y_h^r$$

  $Y_h^r$: space of continuous $\mathbb{P}_r$ finite element functions vanishing on the boundary

- Monte Carlo approximation of expectation: $\omega_i \overset{\text{iid}}{\sim} \mathcal{P},\ i = 1,\dots N$

$$J(u) = \mathbb{E}_\omega[f(u,\omega)] \approx \frac{1}{N}\sum_{i=1}^N f(u,\omega_i)$$

# Finite dimensional approximation

- Finite Element approximation of the PDE: $\forall u \in L^2(D),\ \omega \in \Omega$

$$u \mapsto y_\omega^h(u) \quad \text{solves} \quad \int_D a(\cdot,\omega)\nabla y_\omega^h(u)\cdot\nabla v_h = \int_D (g+u)v_h, \quad \forall v_h \in Y_h^r$$

  $Y_h^r$: space of continuous $\mathbb{P}_r$ finite element functions vanishing on the boundary

- Monte Carlo approximation of expectation: $\omega_i \overset{\text{iid}}{\sim} \mathcal{P},\ i=1,\dots N$

$$J(u) = \mathbb{E}_\omega[f(u,\omega)] \approx \frac{1}{N}\sum_{i=1}^N f(u,\omega_i)$$

- Discrete optimal control problem:

$$\min_{u\in L^2(D)} J^{h,N}(u) := \frac{1}{N}\sum_{i=1}^N f^h(u,\omega_i) = \frac{1}{N}\sum_{i=1}^N \left[\frac{1}{2}\|y_{\omega_i}^h(u) - y_{target}\|^2 + \frac{\beta}{2}\|u\|^2\right]$$

# Finite dimensional approximation

- Finite Element approximation of the PDE: $\forall u \in L^2(D),\ \omega \in \Omega$

$$u \mapsto y_\omega^h(u) \quad \text{solves} \quad \int_D a(\cdot,\omega)\nabla y_\omega^h(u)\cdot\nabla v_h = \int_D (g+u)v_h, \quad \forall v_h \in Y_h^r$$

$Y_h^r$: space of continuous $\mathbb{P}_r$ finite element functions vanishing on the boundary

- Monte Carlo approximation of expectation: $\omega_i \overset{\text{iid}}{\sim} \mathcal{P},\ i = 1,\dots N$

$$J(u) = \mathbb{E}_\omega[f(u,\omega)] \approx \frac{1}{N}\sum_{i=1}^N f(u,\omega_i)$$

- Discrete optimal control problem:

$$\min_{u\in Y_h^r} \quad J^{h,N}(u) := \frac{1}{N}\sum_{i=1}^N f^h(u,\omega_i) = \frac{1}{N}\sum_{i=1}^N \left[\frac{1}{2}\|y_{\omega_i}^h(u) - y_{target}\|^2 + \frac{\beta}{2}\|u\|^2\right]$$

Remark: The unique minimizer $u_\star^{h,N} \in Y_h^r$

# Optimality conditions

primal pbs: $\int_D a(\cdot,\omega_i)\nabla y^h_{\omega_i}\cdot\nabla v_h = \int_D (g+u^{h,N})v_h, \quad \forall v_h\in Y^r_h,\ i=1,\dots,N,$

adjoint pbs: $\int_D a(\cdot,\omega_i)\nabla v_h\cdot\nabla p^h_{\omega_i} = \int_D (y^h_{\omega_i}-y_{tar})v_h, \quad \forall v_h\in Y^r_h,\ i=1,\dots,N,$

sensitivity: $\int_D (\beta u^{h,N}+\frac{1}{N}\sum_{i=1}^{N} p^h_{\omega_i})v_h = 0 \quad \forall v_h\in Y^r_h.$

# Optimality conditions

primal pbs: $$\int_D a(\cdot,\omega_i)\nabla y^h_{\omega_i}\cdot\nabla v_h = \int_D (g+u^{h,N})v_h, \quad \forall v_h\in Y^r_h,\ i=1,\dots,N,$$

adjoint pbs: $$\int_D a(\cdot,\omega_i)\nabla v_h\cdot\nabla p^h_{\omega_i} = \int_D (y^h_{\omega_i}-y_{tar})v_h, \quad \forall v_h\in Y^r_h,\ i=1,\dots,N,$$

sensitivity: $$\int_D (\beta u^{h,N}+\frac{1}{N}\sum_{i=1}^N p^h_{\omega_i})v_h = 0 \qquad \forall v_h\in Y^r_h.$$

## Algebraic system

$$\left[\begin{array}{ccc|ccc|c} A_1 & & & & & & -M \\ & \ddots & & & & & \vdots \\ & & A_N & & & & -M \\ \hline -M & & & A_1^T & & & \\ & \ddots & & & \ddots & & \\ & & -M & & & A_N^T & \\ \hline & & & M & \cdots & M & \beta NM \end{array}\right] \left[\begin{array}{c} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \\ \hline \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \\ \hline \mathbf{u} \end{array}\right] = \left[\begin{array}{c} \mathbf{g} \\ \vdots \\ \mathbf{g} \\ \hline -\mathbf{y}_{tar} \\ \vdots \\ -\mathbf{y}_{tar} \\ \hline \mathbf{0} \end{array}\right]$$

# Outline

1. Problem setting – quadratic optimal control problem
2. Discretization by finite elements + Monte Carlo
3. Deterministic (CG) iterative solvers versus Stochastic Gradient
4. Multilevel stochastic gradient algorithms
5. Conclusions

# Reduced algebraic system

- Several appraoches can be used to solve this coupled system
  [Kouri-Heinkenschloss-eal 2013], [VanBarel-Vandewalle 2018], [Borzì-vonWinckel 2011]

# Reduced algebraic system

- Several appraoches can be used to solve this coupled system
  [Kouri-Heinkenschloss-eal 2013], [VanBarel-Vandewalle 2018], [Borzì-vonWinckel 2011]
- Eliminating $(\mathbf{y}_1, \ldots, \mathbf{y}_N)$ and $(\mathbf{p}_1, \ldots, \mathbf{p}_N)$ and introducing the block matrices

$$\mathcal{A} = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_N \end{bmatrix}, \quad \mathcal{M} = \begin{bmatrix} M & & \\ & \ddots & \\ & & M \end{bmatrix}, \quad \mathbb{1} = \begin{bmatrix} Id \\ \vdots \\ Id \end{bmatrix}$$

leads to a reduced system $\mathcal{G}\mathbf{u} = \boldsymbol{\xi}$ with matrix

$$\mathcal{G} = \beta M + \frac{1}{N} \mathbb{1}^T \mathcal{M} \mathcal{A}^{-T} \mathcal{M} \mathcal{A}^{-1} \mathcal{M} \mathbb{1}$$

EPFL

# Reduced algebraic system

- Several appraoches can be used to solve this coupled system
  [Kouri-Heinkenschloss-eal 2013], [VanBarel-Vandewalle 2018], [Borzì-vonWinckel 2011]
- Eliminating $(\mathbf{y}_1, \ldots, \mathbf{y}_N)$ and $(\mathbf{p}_1, \ldots, \mathbf{p}_N)$ and introducing the block matrices

$$\mathcal{A} = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_N \end{bmatrix}, \quad \mathcal{M} = \begin{bmatrix} M & & \\ & \ddots & \\ & & M \end{bmatrix}, \quad \mathbb{1} = \begin{bmatrix} Id \\ \vdots \\ Id \end{bmatrix}$$

  leads to a reduced system $\mathcal{G}\mathbf{u} = \boldsymbol{\xi}$ with matrix

$$\mathcal{G} = \beta M + \frac{1}{N}\mathbb{1}^T \mathcal{M}\mathcal{A}^{-T}\mathcal{M}\mathcal{A}^{-1}\mathcal{M}\mathbb{1}$$

- The matrix $\mathcal{G}$ is spd and $Cond(\mathcal{G}) = O(\beta^{-1})$ indep. of $h$ and $N$.

# Reduced algebraic system

- Several appraoches can be used to solve this coupled system [Kouri-Heinkenschloss-eal 2013], [VanBarel-Vandewalle 2018], [Borzì-vonWinckel 2011]
- Eliminating $(\mathbf{y}_1, \dots, \mathbf{y}_N)$ and $(\mathbf{p}_1, \dots, \mathbf{p}_N)$ and introducing the block matrices

$$\mathcal{A} = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_N \end{bmatrix}, \quad \mathcal{M} = \begin{bmatrix} M & & \\ & \ddots & \\ & & M \end{bmatrix}, \quad \mathbb{1} = \begin{bmatrix} Id \\ \vdots \\ Id \end{bmatrix}$$

  leads to a reduced system $\mathcal{G}\mathbf{u} = \boldsymbol{\xi}$ with matrix

$$\mathcal{G} = \beta M + \frac{1}{N}\mathbb{1}^T \mathcal{M}\mathcal{A}^{-T}\mathcal{M}\mathcal{A}^{-1}\mathcal{M}\mathbb{1}$$

- The matrix $\mathcal{G}$ is spd and $Cond(\mathcal{G}) = O(\beta^{-1})$ indep. of $h$ and $N$.
- Reduced system can be solved efficiently by e.g. conjugate gradient. Denoting $u_j^{h,N}$ the $j$-th iterate

$$\|u_\star^{h,N} - u_j^{h,N}\| \leq C\rho^j, \qquad \rho = \frac{\sqrt{Cond(\mathcal{G})} - 1}{\sqrt{Cond(\mathcal{G})} + 1}$$

# Deterministic approach

Use standard (deterministic) iterative method (e.g. CG) to solve the fully discrete system

- Error splitting assuming smooth solutions $y_\omega(u_\star), p_\omega(u_\star) \in H^{r+1}(D)$

$$\mathbb{E}[\|u_\star - u_j^{h,N}\|^2] \leq \underbrace{C_1 \rho^{2j}}_{\text{CG error}} + \underbrace{\frac{C_2}{N}}_{\text{MC error}} + \underbrace{C_3 h^{2r+2}}_{\text{FE error}}$$

# Deterministic approach

Use standard (deterministic) iterative method (e.g. CG) to solve the fully discrete system

- Error splitting assuming smooth solutions $y_\omega(u_\star), p_\omega(u_\star) \in H^{r+1}(D)$

$$\mathbb{E}[\|u_\star - u_j^{h,N}\|^2] \leq \underbrace{C_1\rho^{2j}}_{\text{CG error}} + \underbrace{\frac{C_2}{N}}_{\text{MC error}} + \underbrace{C_3 h^{2r+2}}_{\text{FE error}}$$

- Cost to compute $u_j^{h,N}$: assume that the cost of solving 1 PDE is $O(h^{-\gamma d})$ (with $\gamma \in (1,3]$)

$$\Longrightarrow \quad \text{Work to compute } u_j^{h,N}: \qquad Work \sim jNh^{-\gamma d}$$

# Deterministic approach

Use standard (deterministic) iterative method (e.g. CG) to solve the fully discrete system

- Error splitting assuming smooth solutions $y_\omega(u_\star), p_\omega(u_\star) \in H^{r+1}(D)$

$$\mathbb{E}[\|u_\star - u_j^{h,N}\|^2] \leq \underbrace{C_1 \rho^{2j}}_{\text{CG error}} + \underbrace{\frac{C_2}{N}}_{\text{MC error}} + \underbrace{C_3 h^{2r+2}}_{\text{FE error}}$$

- Cost to compute $u_j^{h,N}$: assume that the cost of solving 1 PDE is $O(h^{-\gamma d})$ (with $\gamma \in (1,3]$)

$$\Longrightarrow \quad \text{Work to compute } u_j^{h,N}: \qquad Work \sim jNh^{-\gamma d}$$

- Complexity analysis. Balancing errors contributions: $\rho^j \sim N^{-\frac{1}{2}} \sim h^{r+1} \sim tol$

$$Work(tol) \lesssim \underbrace{tol^{-2}}_{\text{MC}} \underbrace{tol^{-\frac{\gamma d}{r+1}}}_{\text{FE}} \underbrace{\log tol^{-1}}_{\text{solver}}$$

# Stochastic gradient (Robbins-Monro)

Instead of introducing upfront the Monte Carlo approximation and then solve the discrete problem by a deterministic iterative solver, we could apply a stochastic gradient method to the continuous problem (non-discrete in probability)

$$\begin{aligned} u_{j+1}^h &= u_j^h - \tau_j \nabla_u f^h(u_j^h, \omega_j) \\ &= (1 - \tau_j \beta) u_j^h - \tau_j p_{\omega_j}^h(u_j^u) \end{aligned} \qquad \omega_j \overset{\text{iid}}{\sim} \mathcal{P}$$

Learning rate $\tau_j = \frac{\tau_0}{j+\alpha}$

# Stochastic gradient (Robbins-Monro)

Instead of introducing upfront the Monte Carlo approximation and then solve the discrete problem by a deterministic iterative solver, we could apply a stochastic gradient method to the continuous problem (non-discrete in probability)

$$\begin{aligned} u^h_{j+1} &= u^h_j - \tau_j \nabla_u f^h(u^h_j, \omega_j) \\ &= (1 - \tau_j \beta) u^h_j - \tau_j p^h_{\omega_j}(u^u_j) \end{aligned} \qquad \omega_j \overset{\text{iid}}{\sim} \mathcal{P}$$

Learning rate $\tau_j = \frac{\tau_0}{j+\alpha}$

**Proposition** [Martin-Nobile-Krumscheid 2018]

Assuming $y_\omega(u_\star), p_\omega(u_\star) \in H^{r+1}(D)$, for any $\alpha \in \mathbb{R}_+$ and $\tau_0 > \frac{1}{2\beta}$ there exist $D_1, D_2 > 0$ independent of $j$ and $h$ s.t.

SG convergence: $\mathbb{E}[\| u^h_\star - u^h_j \|^2] \leq D_1 j^{-1}$

# Stochastic gradient (Robbins-Monro)

Instead of introducing upfront the Monte Carlo approximation and then solve the discrete problem by a deterministic iterative solver, we could apply a stochastic gradient method to the continuous problem (non-discrete in probability)

$$\begin{aligned} u_{j+1}^h &= u_j^h - \tau_j \nabla_u f^h(u_j^h, \omega_j) \\ &= (1 - \tau_j \beta) u_j^h - \tau_j p_{\omega_j}^h(u_j^u) \qquad \omega_j \overset{\text{iid}}{\sim} \mathcal{P} \end{aligned}$$

Learning rate $\quad \tau_j = \frac{\tau_0}{j+\alpha}$

**Proposition** [Martin-Nobile-Krumscheid 2018]

Assuming $y_\omega(u_\star), p_\omega(u_\star) \in H^{r+1}(D)$, for any $\alpha \in \mathbb{R}_+$ and $\tau_0 > \frac{1}{2\beta}$ there exist $D_1, D_2 > 0$ independent of $j$ and $h$ s.t.

SG convergence: $\quad \mathbb{E}[\| u_\star^h - u_j^h \|^2] \leq D_1 j^{-1}$

Error splitting: $\quad \mathbb{E}[\| u_\star - u_j^h \|^2] \leq 2D_1 j^{-1} + D_2 h^{2r+2}$

# Stochastic gradient (Robbins-Monro)

Instead of introducing upfront the Monte Carlo approximation and then solve the discrete problem by a deterministic iterative solver, we could apply a stochastic gradient method to the continuous problem (non-discrete in probability)

$$\begin{aligned} u_{j+1}^h &= u_j^h - \tau_j \nabla_u f^h(u_j^h, \omega_j) \\ &= (1-\tau_j\beta)u_j^h - \tau_j p_{\omega_j}^h(u_j^u) \qquad \omega_j \overset{\text{iid}}{\sim} \mathcal{P} \end{aligned}$$

Learning rate $\tau_j = \frac{\tau_0}{j+\alpha}$

**Proposition** [Martin-Nobile-Krumscheid 2018]

Assuming $y_\omega(u_\star), p_\omega(u_\star) \in H^{r+1}(D)$, for any $\alpha \in \mathbb{R}_+$ and $\tau_0 > \frac{1}{2\beta}$ there exist $D_1, D_2 > 0$ independent of $j$ and $h$ s.t.

SG convergence: $\mathbb{E}[\|u_\star^h - u_j^h\|^2] \leq D_1 j^{-1}$

Error splitting: $\mathbb{E}[\|u_\star - u_j^h\|^2] \leq 2D_1 j^{-1} + D_2 h^{2r+2}$

Complexity: $Work(tol) \lesssim tol^{-2} tol^{-\frac{\gamma d}{r+1}}$ (no log terms !)

# Numerical results

**Optimal control problem**:

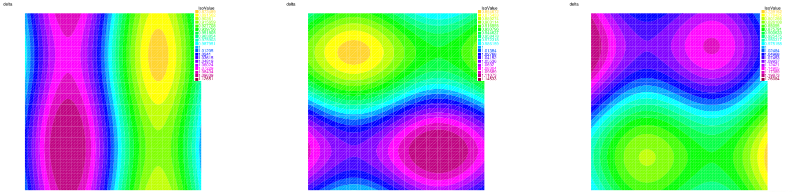
$$\min_{u \in L^2(D)} \frac{1}{2} \mathbb{E}_\omega \left[ \| y_\omega(u) - y_{target} \|^2 \right] + \frac{\beta}{2} \| u \|^2$$

$$\text{subject to} \quad \begin{cases} -\text{div}(a(\cdot,\omega) \nabla y_\omega(u)) = g + u & \text{in } D \\ y_\omega(u) = 0 & \text{on } \partial D \end{cases}$$

**Problem parameters**

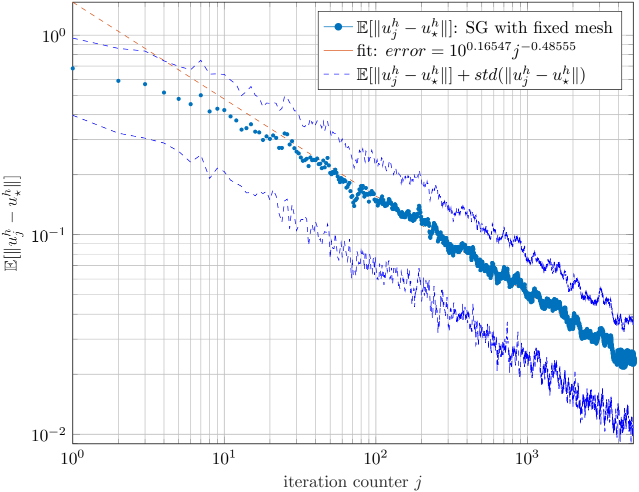
$$D = (0,1)^2, \quad g = 1, \quad y_{target}(x_1, x_2) = \sin(\pi x) \sin(\pi y), \quad \beta = 10^{-4}$$

$$a(x_1, x_2, \boldsymbol{\xi}) = 1 + \exp\{\theta(\xi_1 \cos(1.1\pi x_1) + \xi_2 \cos(1.2\pi x_1) + \xi_3 \sin(1.3\pi x_2) + \xi_4 \sin(1.4\pi x_2))\}$$



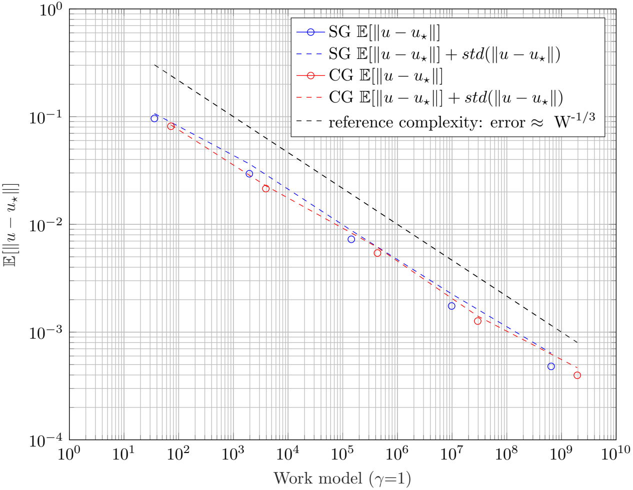

*three realization of the random coefficient*

# Numerical results – SG convergence




*Mean $L^2$ error as a function of iteration counter, estimated by sample average over 100 independent realizations. Fixed mesh size $h = 2^{-4}$ – $\mathbb{P}_1$ finite elements.*

# Numerical results – complexity of CG versus SG




*Complexity plot for CG and SG (average over 20 repetitions)*

# Outline

# Multilevel stochastic gradient

Let $Y^r_{h_0} \subset Y^r_{h_1} \subset \ldots \subset Y^r_{h_L}$ be a sequence of finer and finer FE spaces.

# Multilevel stochastic gradient

Let $Y^r_{h_0} \subset Y^r_{h_1} \subset \ldots \subset Y^r_{h_L}$ be a sequence of finer and finer FE spaces.

**Idea**: In the Stochastic Gradient algorithm, replace the single evaluation $\nabla_u f^h(u_j, \omega_j)$ by a multilevel approx. of the expectation [Heinrich 1998], [Giles 2008]

$$\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j, \cdot)] = \sum_{\ell=0}^{L} \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left[ \nabla_u f^{h_\ell}(u_j, \omega_j^{(i,\ell)}) - \nabla_u f^{h_{\ell-1}}(u_j, \omega_j^{(i,\ell)}) \right]$$

(with the convention $\nabla_u f^{h_{-1}} = 0$) with $\omega_j^{(i,\ell)} \overset{\text{iid}}{\sim} \mathcal{P}$ (drawn independently between levels and at each iteration)

# Multilevel stochastic gradient

Let $Y^r_{h_0} \subset Y^r_{h_1} \subset \ldots \subset Y^r_{h_L}$ be a sequence of finer and finer FE spaces.

**Idea**: In the Stochastic Gradient algorithm, replace the single evaluation $\nabla_u f^h(u_j, \omega_j)$ by a multilevel approx. of the expectation [Heinrich 1998], [Giles 2008]

$$\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j, \cdot)] = \sum_{\ell=0}^{L} \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left[ \nabla_u f^{h_\ell}(u_j, \omega_j^{(i,\ell)}) - \nabla_u f^{h_{\ell-1}}(u_j, \omega_j^{(i,\ell)}) \right]$$

(with the convention $\nabla_u f^{h_{-1}} = 0$) with $\omega_j^{(i,\ell)} \overset{\text{iid}}{\sim} \mathcal{P}$ (drawn independently between levels and at each iteration)

- $L$ controls the bias of the estimator (FE error on level $h_L$)

$$\mathbb{E}\left[\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j, \cdot)] - \mathbb{E}[\nabla_u f(u_j, \cdot)]\right] = \mathbb{E}[\nabla_u f^{h_L}(u_j, \cdot) - \nabla_u f(u_j, \cdot)]$$

# Multilevel stochastic gradient

Let $Y^r_{h_0} \subset Y^r_{h_1} \subset \ldots \subset Y^r_{h_L}$ be a sequence of finer and finer FE spaces.

**Idea**: In the Stochastic Gradient algorithm, replace the single evaluation $\nabla_u f^h(u_j, \omega_j)$ by a multilevel approx. of the expectation [Heinrich 1998], [Giles 2008]

$$\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j, \cdot)] = \sum_{\ell=0}^{L} \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left[ \nabla_u f^{h_\ell}(u_j, \omega_j^{(i,\ell)}) - \nabla_u f^{h_{\ell-1}}(u_j, \omega_j^{(i,\ell)}) \right]$$

(with the convention $\nabla_u f^{h_{-1}} = 0$) with $\omega_j^{(i,\ell)} \overset{\text{iid}}{\sim} \mathcal{P}$ (drawn independently between levels and at each iteration)

- $L$ controls the bias of the estimator (FE error on level $h_L$)

$$\mathbb{E}\left[\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j, \cdot)] - \mathbb{E}[\nabla_u f(u_j, \cdot)]\right] = \mathbb{E}[\nabla_u f^{h_L}(u_j, \cdot) - \nabla_u f(u_j, \cdot)]$$

- $\vec{N} = (N_0, \ldots, N_L)$ controls the variance of the estimator (MC error)

$$\mathbb{V}ar\left[\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j, \cdot)]\right] = \sum_{\ell=0}^{L} \frac{\mathbb{V}ar\left[\nabla_u f^{h_\ell}(u_j, \cdot) - \nabla_u f^{h_{\ell-1}}(u_j, \cdot)\right]}{N_\ell}$$

# Multilevel stochastic gradient algorithm – first version

$$
\begin{aligned}
u_{j+1} &= u_j - \tau_j \mathbb{E}^{MLMC}_{L_j,\vec{N}_j}[\nabla_u f(u_j,\cdot)] \\
&= (1-\tau_j\beta)u_j - \tau_j \sum_{\ell=0}^{L_j} \frac{1}{N_{\ell,j}} \sum_{i=1}^{N_{\ell,j}} \left[ p^{h_\ell}(u_j,\omega_j^{(i,\ell)}) - p^{h_{\ell-1}}(u_j,\omega_j^{(i,\ell)}) \right]
\end{aligned}
$$

Learning rate $\tau_j = \frac{\tau_0}{j+\alpha}$. Notice that $\forall j, \;\; u_{j+1} \in Y^r_{h_{L_j}}$

# Multilevel stochastic gradient algorithm – first version

$$
\begin{aligned}
u_{j+1} &= u_j - \tau_j \mathbb{E}^{MLMC}_{L_j,\vec{N}_j}[\nabla_u f(u_j,\cdot)] \\
&= (1-\tau_j\beta)u_j - \tau_j \sum_{\ell=0}^{L_j} \frac{1}{N_{\ell,j}} \sum_{i=1}^{N_{\ell,j}} \left[ p^{h_\ell}(u_j,\omega_j^{(i,\ell)}) - p^{h_{\ell-1}}(u_j,\omega_j^{(i,\ell)}) \right]
\end{aligned}
$$

Learning rate $\tau_j = \frac{\tau_0}{j+\alpha}$. Notice that $\forall j, \;\; u_{j+1} \in Y^r_{h_{L_j}}$

- We allow $L$ and $\vec{N}$ to depend on the iteration counter $j$. How to choose them optimally ?

# Multilevel stochastic gradient algorithm – first version

$$u_{j+1} = u_j - \tau_j \mathbb{E}^{MLMC}_{L_j,\vec{N}_j}[\nabla_u f(u_j,\cdot)]$$
$$= (1-\tau_j\beta)u_j - \tau_j \sum_{\ell=0}^{L_j} \frac{1}{N_{\ell,j}} \sum_{i=1}^{N_{\ell,j}} \left[p^{h_\ell}(u_j,\omega_j^{(i,\ell)}) - p^{h_{\ell-1}}(u_j,\omega_j^{(i,\ell)})\right]$$

Learning rate $\tau_j = \frac{\tau_0}{j+\alpha}$. Notice that $\forall j,\ \ u_{j+1} \in Y^r_{h_{L_j}}$

- We allow $L$ and $\vec{N}$ to depend on the iteration counter $j$. How to choose them optimally ?
- To recover the optimal control $u_\star$ in the limit, we need $L_j \to \infty$ as $j \to \infty$. On the other hand, $\vec{N}$ does not need to go to $\infty$.

# Multilevel stochastic gradient algorithm – first version

$$u_{j+1} = u_j - \tau_j \mathbb{E}^{MLMC}_{L_j,\vec{N}_j}[\nabla_u f(u_j,\cdot)]$$
$$= (1-\tau_j\beta)u_j - \tau_j \sum_{\ell=0}^{L_j} \frac{1}{N_{\ell,j}} \sum_{i=1}^{N_{\ell,j}} \left[ p^{h_\ell}(u_j,\omega_j^{(i,\ell)}) - p^{h_{\ell-1}}(u_j,\omega_j^{(i,\ell)}) \right]$$

Learning rate $\tau_j = \frac{\tau_0}{j+\alpha}$. Notice that $\forall j, \;\; u_{j+1} \in Y^r_{h_{L_j}}$

- We allow $L$ and $\vec{N}$ to depend on the iteration counter $j$. How to choose them optimally ?
- To recover the optimal control $u_\star$ in the limit, we need $L_j \to \infty$ as $j \to \infty$. On the other hand, $\vec{N}$ does not need to go to $\infty$.

Similar approach proposed in [Dereich-MuellerGronbach 2015] for abstract optimization problem. Different working assumptions but similar results.

# Convergence analysis

- Bias term $\varepsilon_j = \left\| \mathbb{E}[\nabla_u f^{h_{L_j}}(u_\star, \cdot) - \nabla_u f(u_\star, \cdot)] \right\|$
- Variance term $\sigma_j^2 = \mathbb{E}\left[ \left\| \mathbb{E}^{MLMC}_{L_j, \vec{N}_j}[\nabla_u f(u_\star, \cdot)] - \mathbb{E}[\nabla_u f^{h_{L_j}}(u_\star, \cdot)] \right\|^2 \right]$

**Proposition** [Martin-Nobile-Tsilifis 2019]

There exist $\lambda, \mu > 0$ independent of $h$ and $j$ such that

$$\mathbb{E}[\|u_{j+1} - u_\star\|^2] \leq c_j \mathbb{E}[\|u_j - u_\star\|^2] + \lambda \tau_j^2 \sigma_j^2 + \mu \tau_j \varepsilon_j^2$$

with $c_j \sim 1 - \frac{\tau_0 \beta}{j+\alpha}$ $\quad (\Rightarrow \quad \prod_{k=0}^{j} c_k \sim j^{-\tau_0 \beta})$

# Convergence analysis

- Bias term $\varepsilon_j = \left\| \mathbb{E}[\nabla_u f^{h_{L_j}}(u_\star,\cdot) - \nabla_u f(u_\star,\cdot)] \right\|$
- Variance term $\sigma_j^2 = \mathbb{E}\left[ \left\| \mathbb{E}^{MLMC}_{L_j,\vec{N}_j}[\nabla_u f(u_\star,\cdot)] - \mathbb{E}[\nabla_u f^{h_{L_j}}(u_\star,\cdot)] \right\|^2 \right]$

**Proposition** [Martin-Nobile-Tsilifis 2019]

There exist $\lambda, \mu > 0$ independent of $h$ and $j$ such that

$$\mathbb{E}[\|u_{j+1} - u_\star\|^2] \leq c_j \mathbb{E}[\|u_j - u_\star\|^2] + \lambda \tau_j^2 \sigma_j^2 + \mu \tau_j \varepsilon_j^2$$

with $c_j \sim 1 - \frac{\tau_0 \beta}{j+\alpha}$ $\quad (\Rightarrow \ \prod_{k=0}^{j} c_k \sim j^{-\tau_0\beta})$

Optimal error balance: $\tau_j^2 \sigma_j^2 \sim \tau_j \varepsilon_j^2$

# Convergence analysis

- Bias term $\varepsilon_j = \left\| \mathbb{E}[\nabla_u f^{h_{L_j}}(u_\star, \cdot) - \nabla_u f(u_\star, \cdot)] \right\|$
- Variance term $\sigma_j^2 = \mathbb{E}\left[ \left\| \mathbb{E}^{MLMC}_{L_j, \vec{N}_j}[\nabla_u f(u_\star, \cdot)] - \mathbb{E}[\nabla_u f^{h_{L_j}}(u_\star, \cdot)] \right\|^2 \right]$

**Proposition** [Martin-Nobile-Tsilifis 2019]

There exist $\lambda, \mu > 0$ independent of $h$ and $j$ such that

$$\mathbb{E}[\|u_{j+1} - u_\star\|^2] \leq c_j \mathbb{E}[\|u_j - u_\star\|^2] + \lambda \tau_j^2 \sigma_j^2 + \mu \tau_j \varepsilon_j^2$$

with $c_j \sim 1 - \frac{\tau_0 \beta}{j+\alpha}$ $\quad (\Rightarrow \quad \prod_{k=0}^{j} c_k \sim j^{-\tau_0 \beta})$

Optimal error balance: $\tau_j^2 \sigma_j^2 \sim \tau_j \varepsilon_j^2 \quad \Longrightarrow \quad \varepsilon_j^2 \sim \frac{1}{j} \sigma_j^2$

Square bias should decrease faster than variance !

# Compexity analysis: optimal bias/variance decrease

Take $\tau_0 > \frac{1}{\beta}$ and $\eta > 2$

$$\varepsilon_j^2 \sim (j+\alpha)^{1-\eta} \qquad \Longrightarrow \qquad L_j \sim \frac{\eta-1}{2r+2}\log(j+\alpha)$$

$$\sigma_j^2 \sim (j+\alpha)^{2-\eta} \qquad \Longrightarrow \qquad N_{\ell,j} \sim 2^{-\ell\frac{2r+2+\gamma d}{2}}(j+\alpha)^{\eta-2}$$

**Proposition** [Martin-Nobile-Tsilifis 2019]

The work to achieve a mean squared error $\mathbb{E}[\|u_j - u_\star\|^2] = O(tol^2)$ is bounded by

$$Work(tol) \lesssim \begin{cases} tol^{-2}, & 2r+2 > \gamma d, \quad \eta \geq 2 + \frac{\gamma d}{2r+2-\gamma d}, \quad \tau_0 > \frac{\eta-1}{\beta} \\ tol^{-2(1+\frac{1}{\tau_0\beta})}|\log tol|^{3+\frac{1}{\tau_0\beta}}, & 2r+2 = \gamma d, \quad \eta = \tau_0\beta + 1 \\ tol^{-2(\frac{\gamma d}{2r+2}+\frac{1}{\tau_0\beta})}|\log tol|^{\frac{\gamma d}{2r+2}+\frac{1}{\tau_0\beta}}, & 2r+2 < \gamma d, \quad \eta = \tau_0\beta + 1 \end{cases}$$

# Multilevel stochastic gradient algorithm – second version

The multilevel Stochastic Gradient algorithm (MLSG) is in between a Stochastic Gradient and a Full gradient: at each iteration, we compute an approximation $\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j,\cdot)]$ of increasing accuracy ($\varepsilon_j, \sigma_j \to 0$ as $j \to \infty$).

# Multilevel stochastic gradient algorithm – second version

The multilevel Stochastic Gradient algorithm (MLSG) is in between a Stochastic Gradient and a Full gradient: at each iteration, we compute an approximation $\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j,\cdot)]$ of increasing accuracy ($\varepsilon_j, \sigma_j \to 0$ as $j \to \infty$).

Alternative Idea (inspired from "unbiased MLMC estimator" of [Rhee-Glynn 2015]): at each iteration, sample randomly one level $\ell_j$ and one realization $\omega_j$ and compute the difference $\nabla_u f^{h_{\ell_j}}(u_j,\omega_j) - \nabla_u f^{h_{\ell_j-1}}(u_j,\omega_j)$

# Multilevel stochastic gradient algorithm – second version

The multilevel Stochastic Gradient algorithm (MLSG) is in between a Stochastic Gradient and a Full gradient: at each iteration, we compute an approximation $\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j,\cdot)]$ of increasing accuracy ($\varepsilon_j, \sigma_j \to 0$ as $j \to \infty$).

Alternative Idea (inspired from "unbiased MLMC estimator" of [Rhee-Glynn 2015]): at each iteration, sample randomly one level $\ell_j$ and one realization $\omega_j$ and compute the difference $\nabla_u f^{h_{\ell_j}}(u_j,\omega_j) - \nabla_u f^{h_{\ell_j - 1}}(u_j,\omega_j)$

**Randomized Multilevel Stochastic Gradient (RMLSG)** (only for the case $2r+2 > \gamma d$)

$$u_{j+1} = u_j - \tau_j \frac{1}{p_{\ell_j}} \left[ \nabla_u f^{h_{\ell_j}}(u_j,\omega_j) - \nabla_u f^{h_{\ell_j - 1}}(u_j,\omega_j) \right], \quad \ell_j \overset{\text{iid}}{\sim} \{p_\ell\}_{\ell=0}^{L_j}, \;\; \omega_j \overset{\text{iid}}{\sim} \mathcal{P}$$

with $\tau_j = \frac{\tau_0}{j+\alpha}$ and $\{p_\ell \propto 2^{-\ell \frac{2r+2+\gamma d}{2}}\}$ probability mass function on $\{0,\ldots,L_j\}$

# Multilevel stochastic gradient algorithm – second version

The multilevel Stochastic Gradient algorithm (MLSG) is in between a Stochastic Gradient and a Full gradient: at each iteration, we compute an approximation $\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j,\cdot)]$ of increasing accuracy ($\varepsilon_j, \sigma_j \to 0$ as $j \to \infty$).

Alternative Idea (inspired from "unbiased MLMC estimator" of [Rhee-Glynn 2015]): at each iteration, sample randomly one level $\ell_j$ and one realization $\omega_j$ and compute the difference $\nabla_u f^{h_{\ell_j}}(u_j,\omega_j) - \nabla_u f^{h_{\ell_j-1}}(u_j,\omega_j)$

**Randomized Multilevel Stochastic Gradient (RMLSG)** (only for the case $2r+2>\gamma d$)

$$u_{j+1} = u_j - \tau_j \frac{1}{p_{\ell_j}}\left[\nabla_u f^{h_{\ell_j}}(u_j,\omega_j) - \nabla_u f^{h_{\ell_j-1}}(u_j,\omega_j)\right], \quad \ell_j \overset{\text{iid}}{\sim} \{p_\ell\}_{\ell=0}^{L_j}, \;\; \omega_j \overset{\text{iid}}{\sim} \mathcal{P}$$

with $\tau_j = \frac{\tau_0}{j+\alpha}$ and $\{p_\ell \propto 2^{-\ell\frac{2r+2+\gamma d}{2}}\}$ probability mass function on $\{0,\dots,L_j\}$

- Bias: $\varepsilon_j = \|\mathbb{E}[\nabla_u f^{h_{L_j}}(u_j,\cdot) - \nabla_u f(u_j,\cdot)]\|$ (same as for MLSG)

# Multilevel stochastic gradient algorithm – second version

The multilevel Stochastic Gradient algorithm (MLSG) is in between a Stochastic Gradient and a Full gradient: at each iteration, we compute an approximation $\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j,\cdot)]$ of increasing accuracy ($\varepsilon_j, \sigma_j \to 0$ as $j \to \infty$).

Alternative Idea (inspired from "unbiased MLMC estimator" of [Rhee-Glynn 2015]): at each iteration, sample randomly one level $\ell_j$ and one realization $\omega_j$ and compute the difference $\nabla_u f^{h_{\ell_j}}(u_j, \omega_j) - \nabla_u f^{h_{\ell_j-1}}(u_j, \omega_j)$

**Randomized Multilevel Stochastic Gradient (RMLSG)** (only for the case $2r + 2 > \gamma d$)

$$u_{j+1} = u_j - \tau_j \frac{1}{p_{\ell_j}} \left[\nabla_u f^{h_{\ell_j}}(u_j, \omega_j) - \nabla_u f^{h_{\ell_j-1}}(u_j, \omega_j)\right], \quad \ell_j \overset{\text{iid}}{\sim} \{p_\ell\}_{\ell=0}^{L_j}, \;\; \omega_j \overset{\text{iid}}{\sim} \mathcal{P}$$

with $\tau_j = \frac{\tau_0}{j+\alpha}$ and $\{p_\ell \propto 2^{-\ell\frac{2r+2+\gamma d}{2}}\}$ probability mass function on $\{0, \dots, L_j\}$

- Bias: $\varepsilon_j = \|\mathbb{E}[\nabla_u f^{h_{L_j}}(u_j,\cdot) - \nabla_u f(u_j,\cdot)]\|$ (same as for MLSG)
- If one takes $L_j = \infty$, $\forall j$, then the estimator is unbiadsed. Unfortunately, we were not able to prove convergence with this setting.

# Multilevel stochastic gradient algorithm – second version

The multilevel Stochastic Gradient algorithm (MLSG) is in between a Stochastic Gradient and a Full gradient: at each iteration, we compute an approximation $\mathbb{E}^{MLMC}_{L,\vec{N}}[\nabla_u f(u_j,\cdot)]$ of increasing accuracy ($\varepsilon_j, \sigma_j \to 0$ as $j \to \infty$).

Alternative Idea (inspired from "unbiased MLMC estimator" of [Rhee-Glynn 2015]): at each iteration, sample randomly one level $\ell_j$ and one realization $\omega_j$ and compute the difference $\nabla_u f^{h_{\ell_j}}(u_j,\omega_j) - \nabla_u f^{h_{\ell_j-1}}(u_j,\omega_j)$

**Randomized Multilevel Stochastic Gradient (RMLSG)** (only for the case $2r+2>\gamma d$)

$$u_{j+1} = u_j - \tau_j \frac{1}{p_{\ell_j}} \left[ \nabla_u f^{h_{\ell_j}}(u_j,\omega_j) - \nabla_u f^{h_{\ell_j-1}}(u_j,\omega_j) \right], \quad \ell_j \overset{\text{iid}}{\sim} \{p_\ell\}_{\ell=0}^{L_j}, \;\; \omega_j \overset{\text{iid}}{\sim} \mathcal{P}$$

with $\tau_j = \frac{\tau_0}{j+\alpha}$ and $\{p_\ell \propto 2^{-\ell\frac{2r+2+\gamma d}{2}}\}$ probability mass function on $\{0,\ldots,L_j\}$

- Bias: $\varepsilon_j = \|\mathbb{E}[\nabla_u f^{h_{L_j}}(u_j,\cdot) - \nabla_u f(u_j,\cdot)]\|$ (same as for MLSG)
- If one takes $L_j = \infty$, $\forall j$, then the estimator is unbiadsed. Unfortunately, we were not able to prove convergence with this setting.
- Alternatively, decrease the bias as $\varepsilon_j^2 \sim (j+\alpha)^{1-\eta}$

# Compexity analysis of RMLSG: optimal bias decrease

**Proposition** [Martin-Nobile-Tsilifis 2019]

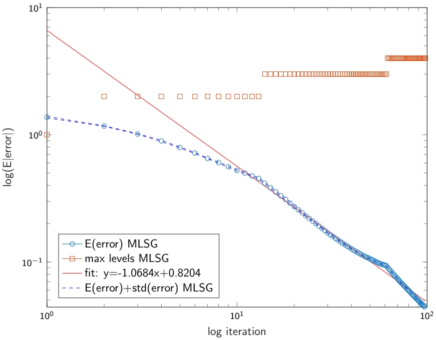
Assume $2r + 2 > \gamma d$ and take $\eta = 2$, $\tau_0 > \frac{1}{\beta}$. Then, the expected work to achieve a mean squared error $\mathbb{E}[\|u_j - u_\star\|^2] = O(tol^2)$ is

$$\mathbb{E}[Work(tol)] \lesssim tol^{-2}$$

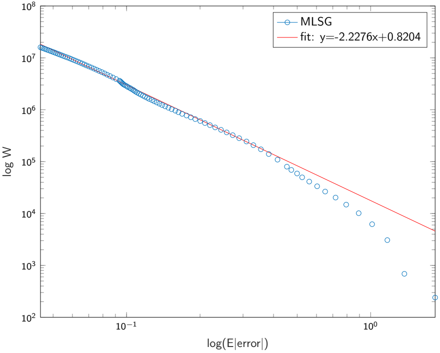
and

$$\frac{\sqrt{\mathbb{V}ar\left[Work(tol)\right]}}{\mathbb{E}\left[Work(tol)\right]} \lesssim tol^{\frac{3(2r+2)-\gamma d}{2(2r+2)}} \xrightarrow{tol \to 0} 0$$

# Numerical results – MLSG complexity



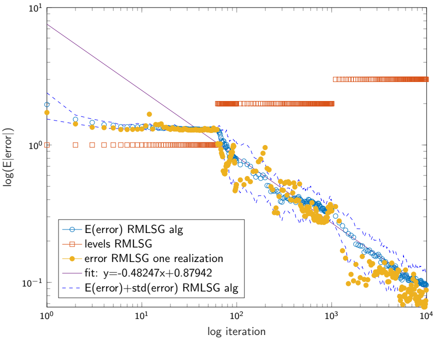

*Mean error vs iteration counter*



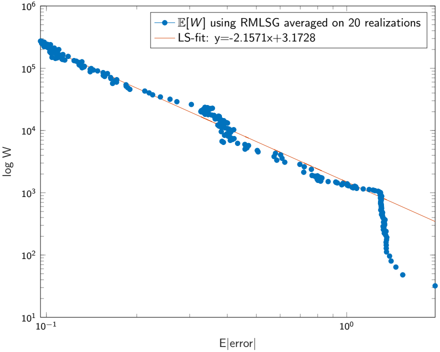

*Cost W versus mean error*

(average over 10 repetitions)

# Numerical results – RMLSG complexity




*Mean error vs iteration counter*




*Cost W versus mean error*

(average over 20 repetitions)

# Outline

1. Problem setting – quadratic optimal control problem
2. Discretization by finite elements + Monte Carlo
3. Deterministic (CG) iterative solvers versus Stochastic Gradient
4. Multilevel stochastic gradient algorithms
5. Conclusions

# Conclusions and future work

- We have analyzed and compared several algorithms to solve a PDE constrained quadratic optimal control problem. Our outcome is that, in the context of a Monte Carlo approximation, the Stochastic gradient algorithm has a slighly better complexity than a deterministic solver such as CG. The multilevel versions, on the other hand, has a substantially better complexity.

# Conclusions and future work

- We have analyzed and compared several algorithms to solve a PDE constrained quadratic optimal control problem. Our outcome is that, in the context of a Monte Carlo approximation, the Stochastic gradient algorithm has a slighly better complexity than a deterministic solver such as CG. The multilevel versions, on the other hand, has a substantially better complexity.
- Our analysis of SG and MLSG uses only the (uniform) Lipschitz and strong convexity properties of the random cost functional. As such, our results may generalize to ortehr problems as e.g.

# Conclusions and future work

- We have analyzed and compared several algorithms to solve a PDE constrained quadratic optimal control problem. Our outcome is that, in the context of a Monte Carlo approximation, the Stochastic gradient algorithm has a slighly better complexity than a deterministic solver such as CG. The multilevel versions, on the other hand, has a substantially better complexity.
- Our analysis of SG and MLSG uses only the (uniform) Lipschitz and strong convexity properties of the random cost functional. As such, our results may generalize to ortehr problems as e.g.
  - general linear second order elliptic equations with random coefficients

# Conclusions and future work

- We have analyzed and compared several algorithms to solve a PDE constrained quadratic optimal control problem. Our outcome is that, in the context of a Monte Carlo approximation, the Stochastic gradient algorithm has a slighly better complexity than a deterministic solver such as CG. The multilevel versions, on the other hand, has a substantially better complexity.
- Our analysis of SG and MLSG uses only the (uniform) Lipschitz and strong convexity properties of the random cost functional. As such, our results may generalize to ortehr problems as e.g.
  - general linear second order elliptic equations with random coefficients
  - boundary control

# Conclusions and future work

- We have analyzed and compared several algorithms to solve a PDE constrained quadratic optimal control problem. Our outcome is that, in the context of a Monte Carlo approximation, the Stochastic gradient algorithm has a slighly better complexity than a deterministic solver such as CG. The multilevel versions, on the other hand, has a substantially better complexity.
- Our analysis of SG and MLSG uses only the (uniform) Lipschitz and strong convexity properties of the random cost functional. As such, our results may generalize to ortehr problems as e.g.
  - general linear second order elliptic equations with random coefficients
  - boundary control
  - non-linear (convex and Lipschitz) cost functionals

# Conclusions and future work

- We have analyzed and compared several algorithms to solve a PDE constrained quadratic optimal control problem. Our outcome is that, in the context of a Monte Carlo approximation, the Stochastic gradient algorithm has a slighly better complexity than a deterministic solver such as CG. The multilevel versions, on the other hand, has a substantially better complexity.
- Our analysis of SG and MLSG uses only the (uniform) Lipschitz and strong convexity properties of the random cost functional. As such, our results may generalize to ortehr problems as e.g.
  - general linear second order elliptic equations with random coefficients
  - boundary control
  - non-linear (convex and Lipschitz) cost functionals
- Extension of the results to more involved risk measures (other than expectation) is in progress.

# Conclusions and future work

- We have analyzed and compared several algorithms to solve a PDE constrained quadratic optimal control problem. Our outcome is that, in the context of a Monte Carlo approximation, the Stochastic gradient algorithm has a slighly better complexity than a deterministic solver such as CG. The multilevel versions, on the other hand, has a substantially better complexity.
- Our analysis of SG and MLSG uses only the (uniform) Lipschitz and strong convexity properties of the random cost functional. As such, our results may generalize to ortehr problems as e.g.
  - general linear second order elliptic equations with random coefficients
  - boundary control
  - non-linear (convex and Lipschitz) cost functionals
- Extension of the results to more involved risk measures (other than expectation) is in progress.
- In the case on only few random variables, a deterministic quadrature (e.g. full tensor or sparse quadrature) may be more accurate than Monte Carlo. In this case, Stochastic Gradient algorithms can still be effective if properly modified (see SAG / SAGA versions).
  Analysis available in [Martin-Nobile 2018].

# Thank you for your attention!

Martin, M.C.
Stochastic approximation methods for PDE constrained optimal control problems with uncertain parameters.
PhD thesis 7233, March 2019, EPFL. doi.org/10.5075/epfl-thesis-7233

Martin, M. and Krumscheid, S. and Nobile F.
Analysis of stochastic gradient methods for PDE-constrained optimal control problems with uncertain parameters
MATHICSE Technical Report 04.2018. doi.org/10.5075/epfl-MATHICSE-263568

Martin, M. and Nobile F. and Tsilifis, P.
Multilevel stochastic gradient methods for PDE-constrained optimal control problems with uncertain parameters
in preparation

Martin, M and Nobile, F.
PDE-constrained optimal control problems with uncertain parameters using SAGA.
arXiv:1810.13378.