PDE-CONSTRAINED OPTIMAL CONTROL PROBLEMS WITH UNCERTAIN PARAMETERS USING SAGA

M. MARTIN AND F. NOBILE

Abstract. We consider an optimal control problem for an elliptic partial differential equation (PDE) with random coefficients. The control function is a deterministic, distributed forcing term that minimizes an expected quadratic regularized loss functional. We consider a Finite Element discretization of the underlying PDEs and a Gaussian-type quadrature formula to approximate the expected loss.

For the computation of the approximate optimal control, we propose a generalization of the SAGA algorithm [2], a type of Stochastic Gradient algorithm with a fixed-length memory term, which computes at each iteration the gradient of the loss functional in only one quadrature point, randomly chosen from a possibly non-uniform distribution. We provide a full error and complexity analysis of the proposed numerical scheme. In particular we compare the complexity of the generalized SAGA algorithm, with that of the Stochastic Gradient (SG) and the Full Gradient (FG) algorithms, applied to the same discretized optimal control problem. We show that SAGA converges exponentially in the number of iterations as for a FG algorithm and has a similar asymptotic computational complexity. Moreover, it features good pre-asymptotic properties, as shown by our numerical experiments, which makes it appealing in a limited-budget context.

Keywords. PDE constrained optimization, risk-averse optimal control, optimization under uncertainty, PDE with random coefficients, stochastic approximation, stochastic gradient, Monte Carlo, SAG, SAGA, importance sampling

AMS subject classifications. 35Q93, 49M99, 65C05, 65N12, 65N30

Introduction

In this paper we consider a risk averse optimal control problem (OCP) for an elliptic PDE with random diffusion coefficients

\[ u^* \in \arg \min_{u \in U} J(u), \quad J(u) = E_{\omega}[f(u, \omega)] \]

where \( \omega \in \Gamma \) denotes a random elementary event, \( f(u, \omega) = \tilde{f}(y_{\beta}(u, \omega)) \) and \( y_{\beta}(u) \) is the solution of an elliptic PDE \( \mathcal{E}(y_{\beta}(u), \omega) = u \) with some random coefficients. Here the right hand side \( u \) is a deterministic function, in a possibly infinite dimensional space \( U \), that acts as a control so as to minimize the functional \( f(\cdot, \omega) \), in an average sense with respect to (w.r.t.) \( \omega \). In particular, in the setting considered in this work, \( u \mapsto f(u, \omega) \) is strongly convex for any \( \omega \in \Gamma \).

Assuming that the randomness can be parametrized in terms of a small number \( M \) of independent random variables, the expectation appearing in the cost functional \( J(u) \) can be written as a \( M \)-dimensional integral and suitably approximated by a quadrature formula as e.g. a tensorized Gaussian quadrature, leading to an approximate optimal control problem

\[ \tilde{u}^* \in \arg \min_{u \in U} \tilde{J}(u), \quad \tilde{J}(u) = \sum_{j=1}^{n} \zeta_j f(u, \eta_j) \]

where \( \eta_j \) are the quadrature knots and \( \zeta_j \) the quadrature weights with \( \sum_{j=1}^{n} \zeta_j = 1 \). For a given control \( u \), evaluating \( \tilde{J}(u) \) entails the computation of the \( n \) solutions \( \{y_{\eta_j}(u)\}_{j=1}^{n} \) of the underlying PDE. This approach is known in the literature as stochastic collocation method and has been analyzed e.g. in [1]. It leads, in favorable cases, to an error in the functional that converges to zero (sub)-exponentially in \( n \), although typically exposed to the curse of dimensionality, hence acceptable only for a small number of random variables. By replacing the tensorized quadrature by a suitable sparse one (see e.g. [3] [11]), dimension free convergence rates have been demonstrated in certain cases (see e.g. [6] [7] [14] [19] and references therein). However, in this work, we stick to the simpler setting of a tensorized Gaussian quadrature formula and a small number of random variables.

To solve the approximate OCP [2], we could consider a steepest descent hereafter called Full Gradient (FG), or a Conjugate Gradient (CG) method which would converge exponentially fast in the number of iterations, i.e. \( \|\tilde{u}_k - \tilde{u}^*\| \leq C \rho^k \) for some \( \rho \in (0,1) \) where \( \tilde{u}_k \) is the \( k \)-th iterate of the method and the error is measured in a suitable norm. The practical limitation of this approach is that each iteration requires the evaluation of a descent direction for \( \tilde{J}(u) \), which entails \( n \) solutions of the PDE and \( n \) solutions of the corresponding adjoint problem. If \( n \) is large, the cost for one single iteration may become excessively high.
A popular technique in the machine learning community to solve optimization problems of the form \( (2) \) is the \textit{Stochastic Gradient} (SG) method \( (13) \) which reads

\[
\hat{u}_{k+1} = \hat{u}_k - \tau_k \zeta_i \nabla f(u, \eta_i)
\]

where the gradient of only one term in the sum is evaluated at each iteration (corresponding to one primal and one adjoint computation) for a randomly drawn indice \( i_k \), and the convergence is achieved by reducing the step size \( \tau_k \) over the iterations. This makes the cost of each iteration affordable. The convergence of the SG method for a PDE-constrained optimal control problem with uncertain parameters has been studied in the recent work \( 10 \) in the context of a Monte Carlo approximation of the expectation appearing in \( 1 \). In particular, we have shown that the root mean squared error \( \mathbb{E}[\|\hat{u}_k - \hat{u}\|^2] \) of the SG method converges with order \( 1/\sqrt{k} \), which is the same order of the Monte Carlo “quadrature” error, and leads to an optimal strategy and a slightly better overall complexity than a FG (or CG) approach. In the setting of this paper, however, the quadrature error decays (sub)-exponentially, and the convergence rate of \( 1/\sqrt{k} \) of the SG method would lead to a much worse complexity than a FG or a CG method.

In recent years, variants of the SG method for a finite dimensional optimization problem of the form \( (2) \), such as the \textit{Stochastic Averaged Gradient} (SAG) method \( 15 \) and the SAGA method \( 4 \) have been proposed, which recover an exponential convergence in the number of iterations, by introducing a memory term which stores all previously computed gradients in the sum and overwrite a term if the corresponding indice is re-drawn. The method presented in \( 4 \) is applicable to the case of uniform weights \( \zeta_j = \frac{1}{n}, i = 1, \ldots, n \) and uniformly drawn indices \( i_k \) over \( \{1, \ldots, n\} \). A variant of the SAGA method that uses a non-uniform sampling of the indices \( i_k \) has been proposed in \( 16 \).

In this paper, we extend the SAGA algorithm to the infinite dimensional setting of problem \( (2) \) and to the case of non-uniform weights, as they appear naturally in a Gaussian quadrature formula. In particular, we propose an importance sampling strategy where the indices \( i_k \) are drawn from a possibly non-uniform distribution, also different from the distribution induced by the weights \( \{\zeta_j\}_{j=1}^n \).

Following similar steps as in \( 4, 16 \), we present a full theoretical convergence analysis of the generalized SAGA method for the infinite dimensional OCP \( (2) \). In particular we show that, asymptotically in \( n \), the optimal sampling measure for the indices \( i_k \) is the \textit{uniform} measure.

We also present a complexity analysis, in terms of computational cost versus accuracy, of the generalized SAGA method to solve the original OCP \( (1) \), which accounts for both the Stochastic Collocation quadrature error as well as the error in solving the primal and adjoint PDEs approximately by the finite element method. The complexity of SAGA is then compared to the complexity of FG as well as SG. Our theoretical results show that the generalized SAGA method has the same asymptotic complexity as the FG method and outperforms SG.

As shown by our numerical experiments, the interest in using SAGA versus FG is in the pre-asymptotic regime, as SAGA often delivers acceptable solutions, from a practical point of view, well before performing \( n \) iterations, i.e. with far less that \( 2n \) PDE solves (we recall that one single FG iteration entails already \( 2n \) PDE solves). In a context of limited budget, SAGA represents therefore a very appealing option.

As pointed out above, in this work we have restricted our study to the case of a small number of random variables and a tensorized Gaussian type quadrature formula, the main constraint in our analysis being that we need positive weights \( \{\zeta_j\} \). This leaves open the question if the methodology can be extended and applied also with other quadrature formulas, such as sparse grid quadratures whose weights are not all positive, and possibly a large number of random variables. This question needs further investigation. We believe, however, that the current work provides an important and necessary step toward further generalizations.

1. Problem setting

We start by introducing the primal problem that will be part of the OCP discussed in the following. Specifically, we consider the problem of finding the solution \( y : D \times \Gamma \to \mathbb{R} \) of the elliptic random PDE

\[
\begin{align*}
-\text{div}(a(x, \omega)\nabla y(x, \omega)) &= g(x) + u(x), & x &\in D, & \omega &\in \Gamma, \\
y(x, \omega) &= 0, & x &\in \partial D, & \omega &\in \Gamma,
\end{align*}
\]

where \( D \subset \mathbb{R}^d \) denotes the physical domain and \( (\Gamma, \mathcal{F}, \mathbb{P}) \) is a complete probability space. The diffusion coefficient \( a \) is a random field, \( g \) is a deterministic source term and \( u \) is the deterministic control. The solution of \( (3) \) for a given control \( u \) will be equivalently denoted \( y_u(\omega) \), or simply \( y(u) \) in what follows. Let \( U = L^2(D) \) be the set of all admissible control functions and \( Y = H_0^1(D) \) the space of the solutions of \( (3) \), then the goal is to determine the optimal control \( u^* \), in the sense that:

\[
u^* \in \arg \min_{u \in U} J(u), \quad \text{s.t.} \quad y_u(\omega) \in Y \quad \text{solves} \quad (3) \quad \text{almost surely (a.s.) in} \ \Gamma.
\]
Here, \( J(u) := \mathbb{E}[f(u, \omega)] \) is the objective function with \( f(u, \omega) = \frac{1}{2}\|y_u(u) - z_d\|^2 + \frac{\beta}{2}\|u\|^2 \) and \( z_d \) is the target function that we would like the state \( y \) to approach as close as possible. We have denoted by \( \| \cdot \| \) the \( L^2(D) \)-norm induced by the inner product \( \langle \cdot, \cdot \rangle \).

1.1. Existence and uniqueness result. We use results from \([10]\) Sections 2 and 3. We recall the three assumptions from \([10]\) that guarantee well posedness of \((4)\) and regularity of solutions.

**Assumption 1.** The diffusion coefficient \( a \in L^\infty(D \times \Gamma) \) is bounded and bounded away from zero a.e. in \( D \times \Gamma \), i.e.

\[ \exists \ a_{\min}, a_{\max} \in \mathbb{R} \text{ such that } 0 < a_{\min} \leq a(x, \omega) \leq a_{\max} \text{ a.e. in } D \times \Gamma. \]

**Assumption 2.** The regularization parameter \( \beta \) is strictly positive, i.e. \( \beta > 0 \) and the deterministic source term is such that \( g \in L^2(D) \).

In what follow, we denote the \( L^2(D) \)-functional representation of the Gateaux derivative of \( J \), by \( \nabla_u J(u) \), namely

\[
\int_D \nabla_u J(u) \delta u \, dx = \lim_{\epsilon \to 0} \frac{J(u + \epsilon \delta u) - J(u)}{\epsilon} \quad \forall \delta u \in L^2(D).
\]

Existence and uniqueness of the OCP \((4)\) can be stated as follows.

**Theorem 1.** Under Assumptions 1 and 3 the OCP \((4)\) admits a unique control \( u^* \in U \). Moreover

\[
\nabla_u J(u) = \beta u + \mathbb{E}[p_u(u)],
\]

where \( p_u(u) = p \) is the solution of the adjoint problem (a.s. in \( \Gamma \))

\[
\left\{
\begin{array}{l}
- \text{div}(a(\cdot, \omega) \nabla p(\cdot, \omega)) = \frac{g}{\beta} \text{ in } D, \\
p(\cdot, \omega) = 0 \text{ on } \partial D.
\end{array}
\right.
\]

We recall as well the weak formulation of \((3)\), which reads

\[
\text{find } y_u \in Y \text{ s.t. } b_u(y_u, v) = (g + u, v) \quad \forall v \in Y \quad \text{ for a.e. } \omega \in \Gamma,
\]

where \( b_u(y, v) := \int_D a(\cdot, \omega) \nabla y \nabla v \, dx \). Similarly, the weak form of the adjoint problem \((5)\) reads:

\[
\text{find } p_u \in Y \text{ s.t. } b_u(v, p_u) = \langle v, y_u - z_d \rangle \quad \forall v \in Y \quad \text{ for a.e. } \omega \in \Gamma.
\]

We can thus rewrite the OCP \((4)\) equivalently as:

\[
\left\{
\begin{array}{l}
\min_{u \in U} J(u), \quad J(u) = \frac{1}{2} \mathbb{E}[\|y_u(u) - z_d\|^2] + \frac{\beta}{2} \|u\|^2 \\
\text{s.t. } y_u(u) \in Y \text{ solves } b_u(y_u(u), v) = (g + u, v) \quad \forall v \in Y \quad \text{ for a.e. } \omega \in \Gamma.
\end{array}
\right.
\]

We continue recalling two regularity results, that have been proven in \([10]\), about Lipschitz property and strong convexity for \( f \) in the particular setting of the problem considered here.

**Lemma 1** (Lipschitz condition). The random functional \( f(u, \omega) \) is such that:

\[
\|\nabla_u f(u_1, \omega) - \nabla_u f(u_2, \omega)\| \leq L\|u_1 - u_2\| \quad \forall u_1, u_2 \in U \text{ and a.e. } \omega \in \Gamma,
\]

with \( L = \beta + \frac{C_p}{a_{\min}} \), where \( C_p \) is the Poincaré constant, \( C_p = \sup_{v \in \mathbb{X}} \frac{\|v\|}{\|\nabla v\|} \).

**Lemma 2** (Strong Convexity). The (random) functional \( f(u, \omega) \) is such that:

\[
\frac{1}{2} \|u_1 - u_2\|^2 \leq \langle \nabla_u f(u_1, \omega) - \nabla_u f(u_2, \omega), u_1 - u_2 \rangle \quad \forall u_1, u_2 \in U \text{ and a.e. } \omega \in \Gamma,
\]

with \( l = 2\beta \).

1.2. Finite Element approximation. In order to compute numerically an optimal control we consider a Finite Element (FE) approximation of the infinite dimensional OCP \((4)\). Let us denote by \( \{\tau_h\}_{h > 0} \) a family of regular triangulation of \( D \) and choose \( Y^h \) to be the space of continuous piece-wise polynomial functions of degree \( r \) over \( \tau_h \) that vanish on \( \partial D \), i.e. \( Y^h = \{y \in C^0(\bar{D}) : y_K \in \mathbb{P}_r(K) \quad \forall K \in \tau_h, y|_{\partial D} = 0 \} \subset Y, \) and \( U^h = Y^h \). We reformulate the OCP \((4)\) as a finite dimensional OCP in the FE space:

\[
\left\{
\begin{array}{l}
\min_{h \in U^h} J^h(u^h), \quad J^h(u^h) = \frac{1}{2} \mathbb{E}[\|y_u^h(u^h) - z_d\|^2] + \frac{\beta}{2} \|u^h\|^2 \\
\text{s.t. } y_u^h \in Y^h \text{ and } b_u(y_u^h(u^h), v^h) = \langle u^h + g, v^h \rangle \quad \forall v^h \in Y^h \quad \text{ for a.e. } \omega \in \Gamma.
\end{array}
\right.
\]

Under the following regularity assumption on the domain and diffusion coefficient:

**Assumption 3.** The domain \( D \subset \mathbb{R}^d \) is polygonal convex and the random field \( a \in L^\infty(D \times \Gamma) \) is such that \( \nabla a \in L^\infty(D \times \Gamma) \),
the following error estimate has been obtained in \cite{martin2017}. In order to lighten the notations, we omit the subscript \( \omega \) in \( y_{\omega}() \) and \( p_{\omega}() \) from now on.

**Theorem 2.** Let \( u^* \) be the optimal control, solution of problem \cite{martin2017}, and denote by \( u^{h*} \) the solution of the approximate problem \cite{martin2017}. Suppose that \( y(u^*), p(u^*) \in L^2(\Gamma; H^{r+1}(D)) \) and Assumption \ref{ass:regularity} holds; then

\begin{equation}
\|u^* - u^{h*}\|^2 + E[\|y(u^*) - y^{h}(u^{h*})\|^2] + h^2 E[\|y(u^*) - y^{h}(u^{h*})\|^2_{H^1}]
\leq A_1 h^{2r+2} \{E[\|y(u^*)\|^2_{H^{r+1}}] + E[\|p(u^*)\|^2_{H^{r+1}}]\},
\end{equation}

with a constant \( A_1 \) independent of \( h \).

The next step is to approximate the expectation \( E[\cdot] \) in \cite{martin2017} by a suitable quadrature formula \( \hat{E}[\cdot] \). This is detailed in the next section.

1.3. **Collocation method.** We describe here a semi-discrete (approximation in probability only) OCP obtained by replacing the exact expectation \( E[\cdot] \) in \cite{martin2017} by a suitable quadrature formula \( \hat{E}[\cdot] \). We assume that the random diffusion coefficient can be represented as a function of a finite number of independent uniformly distributed random variables:

\[ a = a(x, \xi) \]

with \( \xi = (\xi_1, \ldots, \xi_M) \) and \( \xi \overset{i.i.d}{\sim} U([-1, 1]) \). Hence, in this case since the whole problem is parameterized by the random vector \( \xi \), we can take a probability space \( \Gamma = [-1, 1]^M \), \( \mathcal{F} = \mathcal{B}(\Gamma) \) the Borel \( \sigma \)-algebra on \( \Gamma \), and \( P(d\xi) = \otimes_{i=1}^M d\xi_i \) the uniform product measure on \( \Gamma \). In this case we chose as a quadrature formula the tensor Gaussian quadrature built on Gauss-Legendre quadrature points. In particular, if \( X : \Gamma \to \mathbb{R} \), \( \xi \mapsto X(\xi) = X(\xi_1, \ldots, \xi_M) \), is a random variable with finite mean, then the Gauss-Legendre quadrature formula is given by

\begin{equation}
\hat{E}[X] = \sum_{j=1}^n \zeta_j X(\eta_j),
\end{equation}

where \( n \) is the total number of points used, \( \{\zeta_j\} \) are the positive quadrature weights and \( \{\eta_j\} \), the associated quadrature knots. The semi-discrete collocation problem then reads:

\begin{align}
\begin{cases}
\min_{\hat{u} \in U} \hat{J}(\hat{u}), & \hat{J}(\hat{u}) = \frac{1}{2} \hat{E}[\|y_{\eta}(\hat{u}) - z_d\|^2] + \frac{\beta}{2} \|\hat{u}\|^2 \\
\quad s.t. \quad y_{\eta}(\hat{u}) \in Y & \text{and} \\
\quad b_{\eta}(y_{\eta}(\hat{u}), v) = (g + \hat{u}, v) & \forall v \in Y \quad j = 1, \ldots, n.
\end{cases}
\end{align}

An error estimate has been shown in \cite{martin2017}.

**Lemma 3.** Let \( u^* \) be the optimal control, solution of \cite{martin2017} and \( \hat{u}^* \) the solution of the semi-discrete OCP \cite{martin2017}. Then there exists \( A_2 > 0 \) s.t.

\begin{equation}
\|u^* - \hat{u}^*\|^2 + \hat{E}[\|y(u^*) - y(\hat{u}^*)\|^2] \leq A_2 \{E[p(u^*)] - \hat{E}[p(\hat{u}^*)]\}^2
\end{equation}

To quantify the convergence rate of the right hand side of \cite{martin2017}, one has first to understand the smoothness of the function \( \xi \mapsto p(u) \) for a generic \( u \in U \). For this, we make the regularity assumption on the diffusion coefficient.

**Assumption 4.** The parametric diffusion coefficient \( \xi \mapsto a(\cdot, \xi) \in L^\infty(D) \) is analytic in each variable \( (\xi_1, \ldots, \xi_M) \) in \( \Gamma \) and there exist \( 0 < \gamma_1, \ldots, \gamma_M \in \mathbb{R} \) and \( A_3 > 0 \) such that

\begin{equation}
\left\| \frac{\partial^k a(\cdot, \xi)}{\partial \xi_j^k} \right\|_{L^\infty(D)} \leq A_3 k! \gamma_j^k
\end{equation}

Then following \cite{martin2017}, it can be shown that for any \( u \in U \), the primal solution \( \xi \mapsto y_\eta(u) \in Y \) and the adjoint solution \( \xi \mapsto p_\eta(u) \in Y \) are both analytic in \( \Gamma \) (see also \cite{martin2017} Lemma 7) and the following result holds:

**Theorem 3.** Denoting by \( \hat{u}^* \) the solution of the semi-discrete (in probability) optimal control problem \cite{martin2017} with \( \hat{E} = E_q^{GL}[\cdot] \) the tensor Gauss-Legendre quadrature formula with \( q = (q_1, \ldots, q_M) \) points in each of the variables \( (\xi_1, \ldots, \xi_M) \), and \( p(\hat{u}^*) \) the corresponding adjoint solution, there exist \( A_4 > 0 \) and \( 0 < s_1, \ldots, s_M \in \mathbb{R} \) independent of \( q \) s.t.

\begin{equation}
\|E[p(\hat{u}^*)] - E_q^{GL}[p(\hat{u}^*)]\|^2 \leq A_4 \sum_{n=1}^M e^{-s_n q_n}.
\end{equation}

Clearly, the discretization in space by Finite Elements \cite{martin2017} and in probability by Gauss-Legendre formula \cite{martin2017} can be combined to obtain the fully discrete OCP:

\begin{align}
\begin{cases}
\min_{\hat{u} \in U} \hat{J}^h(\hat{u}), & \hat{J}^h(\hat{u}) = \frac{1}{2} \hat{E}[\|y^h_{\eta}(\hat{u}) - z_d\|^2] + \frac{\beta}{2} \|\hat{u}\|^2 \\
\quad s.t. \quad y^h_{\eta}(\hat{u}) \in Y^h & \text{and} \\
\quad b^h(\eta, v) = (\hat{u}, v) & \forall v \in Y^h \quad \text{for a.e. } \omega \in \Gamma.
\end{cases}
\end{align}
If \( \hat{u}^{h*} \) denotes the solution of OCP [18], the total error will satisfy

\[
\|u^* - \hat{u}^{h*}\|^2 \leq A_5 \left( h^{2r+2} + \sum_{n=1}^{M} e^{-\kappa n q_n} \right),
\]

for a suitable constant \( A_5 > 0 \) independent of \( h \) and \( \{q_n\} \). The following section is dedicated to optimization techniques used to tackle such optimization problem. In particular, we focus on Stochastic Approximation methods as Stochastic Gradient and Stochastic Average Gradient. To keep the notation light, we present the different optimization algorithms and convergence estimates only for the semi-discrete problem [15], although all results extend straightforwardly to the fully discrete case.

2. Review of Stochastic Approximation methods

We recall two optimization techniques, namely Stochastic Gradient (SG) method, and Stochastic Averaged Gradient (SAG) method, mainly used in machine learning, and well adapted to solve optimization problems whose objective function is the sum of a large number of terms, as in our semi-discrete problem [15]. We consider in this section the general optimization problem

\[
\min_{u \in U} \hat{J}(u), \quad \hat{J}(u) = \frac{1}{n} \sum_{i=1}^{n} g_i(u).
\]

where each function \( g_i \) is convex, differentiable, each gradient \( \nabla g_i \) is Lipschitz-continuous, as defined in [10] replacing \( f \) with \( g_i \), with a common Lipschitz constant \( L \), and \( U \) is finite dimensional.

2.1. Stochastic Gradient (SG). Known in literature as Stochastic Approximation (SA) or Stochastic Gradient (SG) [3, 12, 13, 15, 17], the classic version of such a method, the so-called Robbins-Monro method, works as follows. Within the steepest descent algorithm the exact gradient \( \nabla \hat{J}(u) = \frac{1}{n} \sum_{i=1}^{n} \nabla g_i(u) \) is replaced by one particular term of the sum, \( \nabla g_{i_k}(u) \), where \( i_k \) is chosen at random at each iteration step \( k \) of the optimization algorithm:

\[
-u_{k+1} = u_k - \tau_k \nabla g_{i_k}(u_k).
\]

Here, \( i_k \sim \mathcal{U}\{1, \ldots, n\} \) are iid uniform random variables on \( \{1, \ldots, n\} \). In (21), \( \tau_k \) is the step-size of the algorithm and decreases as \( 1/k \) in the usual approach. For the following theorem, we assume that each \( g_i \) is strongly-convex with a common constant \( l \) defined as in [11] replacing \( f \) with \( g_i \).

**Theorem 4.** Let \( \hat{u}^* \) be the solution of problem (20) and denote by \( u_k \) the \( k \)-th iterate of (21). For the choice \( \tau_k = \tau_0/k \) with \( \tau_0 > 1/l \), then we have:

\[
\mathbb{E}[\|u_k - \hat{u}^*\|^2] \leq A_6 k^{-1},
\]

for a suitable constant \( A_6 > 0 \) independent of \( k \).

2.2. Stochastic Average Gradient (SAG). Another optimization method, called SAG, has recently been introduced in [15]. It relies on the same idea as the SG algorithm, but introduces a memory, which stores the gradients computed using older controls, and averages over them to compute the new gradient direction. The SAG memory-based scheme reads

\[
-u_{k+1} = u_k - \frac{\tau_k}{n} \sum_{j=1}^{n} \nabla g_j(\phi_j^{k+1}),
\]

where at each iteration \( k \) an indice \( i_k \in \{1, \ldots, n\} \) is selected at random, and we set

\[
\phi_j^{k+1} = \begin{cases} u_k & \text{if } j = i_k; \\ \phi_j^k & \text{otherwise}. \end{cases}
\]

Again, the indices \( i_k \sim \mathcal{U}\{1, \ldots, n\} \) are iid uniform random variables. If we assume that each \( g_i \) is strongly-convex with a common constant \( l \) defined as in [11], the authors of [15] have proven that the convergence is exponential in \( k \).

**Theorem 5.** Let \( \hat{u}^* \) be the solution of problem (20) and denote by \( u_k \) the \( k \)-th iterate of (23) with \( \tau_k = \frac{1}{4nL} \). Then

\[
\mathbb{E}[\|u_k - \hat{u}^*\|^2] \leq A_7 \left( 1 - \min \left\{ \frac{L}{16L}, \frac{1}{8n} \right\} \right)^k,
\]

for a suitable constant \( A_7 > 0 \) independent of \( k \).
The objective function needed can dramatically limit this algorithm. The major improvement of this method with respect to SG is its exponential factor (1−\epsilon)k convergence rate for strongly-convex objectives, similar to the full gradient method, versus an algebraic 1/k rate for the SG method. Table 1 summarizes the different convergence rates, for both the objective functional and the control for these two methods.

**Remark 1.** Notice that in SAG, the step-size \( \tau_k \) does not necessarily decrease and remains usually fixed. The factor (1−\epsilon) in the convergence rate in Table 1 depends on the Lipschitz constant \( L \), the strong-convexity constant \( l \), and on the parameter \( n \) such that:

\[
\epsilon = \min \left( \frac{l}{16L}, \frac{1}{8n} \right),
\]

when a fixed constant step-size \( \tau_k = \frac{1}{16L} \) is used (see [15]).

As pointed out in [15], despite the fact that \( n \) appears in the convergence rate of SAG, in the case where \( n > \frac{L}{2\epsilon} \), performing \( n \) effective passes through the quadrature knots, reduces the error by a factor \( (1−1/8n)^n \leq \exp(-1/8) \), which is independent of \( n \). Thus, in this setting, each pass through all the data reduces the error by a constant multiplicative factor as in the FG algorithm.

### 2.3. SAGA.

We recall here also a slightly modified version of SAG, called SAGA, proposed in [4] where the updated part in the gradient estimator is changed by a factor \( n \). It makes the gradient estimator unbiased, and simplifies the proof of convergence. The SAGA iterative scheme reads:

\[
u_{k+1} = u_k - \tau_k \left( \nabla g_{i_k}(u_k) - \nabla g_{i_k}^{\ast}(\phi_{i_k}^{n}) + \frac{1}{n} \sum_{j=1}^{n} \nabla g_{j}(\phi_{j}^{n}) \right)
\]

where, as for SAG, the indices \( i_k \sim U(\{1,\ldots, n\}) \) are drawn independently and \( \phi_{j}^{n} \) is updated as in (24). For comparison, SAG can be rewritten equivalently as:

\[
u_{k+1} = u_k - \tau_k \left( \nabla g_{i_k}(u_k) - \nabla g_{i_k}^{\ast}(\phi_{i_k}^{n}) + \frac{1}{n} \sum_{j=1}^{n} \nabla g_{j}(\phi_{j}^{n}) \right).
\]

The convergence rate of SAGA remains the same as for SAG, while lightening the proof of convergence. In the next section we apply SAGA to the OCP [15] combined with an importance sampling strategy, and extend its convergence proof to this setting.

### 3. Stochastic Approximation methods in the context of PDE constrained OCP

We aim now at applying SG and SAG/SAGA to the semi-discrete OCP [15] (or its fully discrete counterpart). The objective function \( \hat{J}(u) \) in [15] reads:

\[
\hat{J}(u) = \frac{1}{2} \mathbb{E}\|y^{\ast}(u) - z_d\|^2 + \frac{\beta}{2} \|u\|^2
\]

with \( f_i(u) = \frac{1}{2}\|y_{n_i}(u) - z_d\|^2 + \frac{\beta}{2} \|u\|^2 \) and \( \nabla_u f_i(u) = \beta u + p_{n_i}(u) \), where \( \{z_i\} \) are weights of the Gauss Legendre quadrature formula and \( \{n_i\} \) its knots. One possibility to apply SG or SAG/SAGA to the OCP \( \min_{u \in U} \hat{J}(u) \) is to rewrite \( \hat{J}(u) \) as:

\[
\hat{J}(u) = \frac{1}{n} \sum_{i=1}^{n} g_i(u)
\]

with \( g_i(u) = n_i z_i f_i(u) \). However, the functions \( f_i(u) \) are naturally weighted by the non-uniform weights \( \{z_i\} \) and this raises the question whether the indices \( i_k \) in the Stochastic Approximation techniques should be drawn from a uniform or non-uniform distribution. We take the second, more general, approach by introducing a
discrete probability measure \( \tilde{\zeta} \) on \( \{1, \ldots, n\} \), \( \tilde{\zeta}(j) = \tilde{\zeta}_j > 0 \), with \( \sum_{j=1}^n \tilde{\zeta}_j = 1 \) and using an importance sampling strategy.

Hence the modified Stochastic Gradient method with importance sampling for the OCP [15] reads: Similarly

**Algorithm 1:** SG on PDE constrained OCP, with non-uniformly sampled indices

**given** \( u_k \)

**sample** \( i_k \sim \tilde{\zeta} \)

**compute** \( u_{k+1} = u_k - \tau_k \frac{\zeta_{i_k}}{\zeta_{i_k}} \nabla f_{i_k}(u_k) \).

the modified SAGA method with importance sampling for the OCP [15] reads:

**Algorithm 2:** SAGA on PDE constrained OCP, with non-uniformly sampled indices

**given** \( u_k, \{\phi_j^k\}_{j=1}^n \)

**sample** \( i_k \sim \zeta \)

**compute** \( u_{k+1} = u_k - \tau_k \left( (\nabla f_{i_k}(u_k) - \nabla f_{i_k}(\phi_j^k)) \frac{\zeta_{i_k}}{\tilde{\zeta}_{i_k}} + \sum_{j=1}^n \zeta_j \nabla f_j(\phi_j^k) \right) \)

**set** \( \phi_{j+1}^k = \begin{cases} \phi_j^k & \text{if } j = i_k, \\ \phi_j^k & \text{otherwise.} \end{cases} \)

In practice, in the SAGA algorithm, we do not store the past controls \( \phi_j^k \), rather the past gradients \( \text{grad}_j^k = \nabla u f_j(\phi_j^k) \). Similarly, we do not recompute at each iteration \( k \) the whole sum \( G_k = \sum_{j=1}^n \zeta_j \nabla f_j(\phi_j^k) \), rather update it using the formulas

\[ G_{k+1} = G_k - \zeta_{i_{k+1}} \text{grad}_{i_{k+1}} + \zeta_{i_{k+1}} \nabla f_{i_{k+1}}(u_k) \]

and then the memory place \( \text{grad}_{i_{k+1}} \) as:

\[ \text{grad}_{i_{k+1}} = \begin{cases} \nabla f_{i_{k+1}}(u_k) & \text{if } j = i_{k+1}, \\ \text{grad}_j^k & \text{otherwise.} \end{cases} \]

We point out that both the SG and SAGA methods applied to the OCP [15] require 2 PDE’s solved per iteration. Moreover SAGA requires to store 2n PDE solutions at all iterations.

3.1. **Convergence and complexity analysis of the modified SG Algorithm [1]**

Following the analysis in [10], we can bound the Mean Squared Error (MSE) of the SG iterates assuming a weighted summability property on the discrete probability \( \{\tilde{\zeta}_j\}_j \) used to sample the indice \( i_k \) at iteration \( k \):

**Assumption 5** (Weights summability). *Let us define*

\[ \tilde{S}_n = \sum_{j=1}^n \tilde{\zeta}_j^2. \]

*There exist* \( 0 < \tilde{S} < \infty \) *s.t. for every* \( n \in \mathbb{N} \),

\[ \tilde{S}_n \leq \tilde{S}. \]

Then, one can establish the following bound on the MSE when applying SG Algorithm [1]

**Theorem 6.** Denoting by \( \hat{u}_k^M \) the \( k \)-th iteration of the modified Algorithm [1] applied to the fully discrete OCP [15] with FE mesh size \( h \) and Gauss-Legendre quadrature formula with \( (q_1, \ldots, q_M) \) knots in each stochastic direction, then we can bound the MSE, \( E[\|\hat{u}_k^M - u^*\|^2] \), as:

\[ E[\|\hat{u}_k^M - u^*\|^2] \leq B_1 k^{-1} + B_2 \sum_{n=1}^M e^{-\kappa_n q_n} + B_3 h^{2r+2}, \]

with constants \( B_1, B_2, B_3 \) independent of \( h, \{q_n\}_n \) and \( k \).

We omit the proof as it follows very similar steps as in [10]. We now analyze the complexity of the SG algorithm [1] in terms of computational work \( W \) versus accuracy \( \text{tol} \).

**Corollary 1.** *In order to achieve a given tolerance* \( \text{tol} \), *i.e. to guarantee that* \( E[\|\hat{u}_k^M - u^*\|^2] \lesssim \text{tol}^2 \), *the total required computational work is bounded by*

\[ W \lesssim \text{tol}^{-2} \frac{\text{tol}^{2r}}{\gamma}. \]

where we assume that the primal and adjoint problems can be solved, using a triangulation with mesh size \( h \), in computational time \( C_h = O(h^{-d\gamma}) \). Here, \( \gamma \in [1,3] \) is a parameter representing the efficiency of the linear
solver used (e.g. $\gamma = 3$ for a direct solver and $\gamma = 1$ up to a logarithm factor for an optimal multigrid solver), while $d$ is the dimension of the physical space. The memory space required to store the gradient and solution at each iteration scales as

$$
\text{storage} \lesssim \text{tol}^{\frac{d-1}{r}}
$$

Proof. If we want to guarantee an error of order $O(\text{tol})$, we can equalize the three terms on the right hand side of (31) to $\text{tol}$ thus obtaining:

$$
\begin{align*}
    h &= O(\text{tol}^{\frac{1}{r}}), \\
    q_j &= \frac{2}{s_j} \log(\text{tol}^{-1}), \\
    n &= \left(\frac{2}{R} \log(\text{tol}^{-1})\right)^M, \\
    k &= O(\text{tol}^{-2}).
\end{align*}
$$

where we have set $R = \left(\prod_{j=1}^M s_j\right)^{\frac{1}{M}}$ the geometric mean of $(s_1, \ldots, s_M)$. As $W$ denotes the computational work (proportional to time if we don’t use any parallel computing strategy), we have

$$
W = 2C_h k \lesssim \text{tol}^{-2} \text{tol}^{\frac{d-1}{r}}.
$$

The memory space required to store one gradient and one current control $u_k$, is proportional to $h^{-d}$ thus leading to:

$$
\text{storage} \lesssim \text{tol}^{\frac{d-1}{r}}.
$$

3.2. Convergence analysis of the modified SAGA Algorithm We prove in Theorem 7 below the exponential convergence in the number of iterations of Algorithm 2. The outcome of our analysis in that uniform sampling of the index $i_k$ (i.e. $\tilde{\zeta}_j = \frac{1}{n}$, $\forall j = 1, \ldots, n$) is indeed optimal in the sense that it provides the best convergence rate, asymptotically in $n$.

The proof is inspired from [4] and is valid under the assumption that the weights $\{\zeta_j\}_j$ in (29) are positive and sum up to 1, which holds for Gaussian quadrature formulas, each $f_i$ is Lipschitz with the same Lipschitz constant $L$, which is guaranteed for OCP (15) by Lemma 1, and $\tilde{J}(u) = \sum_{i=1}^n \zeta_i f_i(u)$ is strongly convex, which is guaranteed by Lemma 3. In what follows, we denote by $F_k$ the $\sigma$-algebra generated by the random variables $i_0, i_1, \ldots, i_k-1$ and denote by $E[\cdot|F_k]$ the conditional expectation to such $\sigma$-algebra. Moreover, in the remaining of this Section, we use the shorthand notation $f_j^*(u)$ for $\nabla_u f^*_j(u)$. For the convergence proof, we also need to introduce the quantity

$$
Q_k = \sum_{j=1}^n \frac{\zeta_j^2}{\zeta_j} \|f_j^*(\phi_j^k) - f_j^*(\hat{u}^*)\|^2
$$

where $\hat{u}^*$ denotes, as usual, the optimal control, solution of the semi-discrete OCP (15). We start our convergence analysis by few technical Lemmas.

Lemma 4. We have the following bound on the conditional expectation $E[Q_{k+1}|F_k]$:

$$
E[Q_{k+1}|F_k] \leq \max_j (1 - \tilde{\zeta}_j) Q_k + S_n L^2 \|u_k - \hat{u}^*\|^2 \quad \text{with} \quad S_n = \sum_{p=1}^n \zeta_p^2
$$

Proof. We write the conditional expectation as a sum over the possible values $i_k = p$, $p \in \{1, \ldots, n\}$;
\[
E[Q_{k+1}|F_k] = E \left[ \sum_{j=1}^{n} \frac{\zeta_j^2}{\zeta_j} \left| f'_j(\phi_j^k) - f'_j(\tilde{\alpha}^*) \right|^2 | F_k \right]
\]
\[
= \sum_{p=1}^{n} \frac{\zeta_p^2}{\zeta_p} \left( \sum_{j=1}^{n} \left( \frac{\zeta_j^2}{\zeta_j} \left| f'_j(\phi_j^k) - f'_j(\tilde{\alpha}^*) \right|^2 + \frac{\zeta_j^2}{\zeta_p} \left| f'_p(u_k) - f'_p(\tilde{\alpha}^*) \right|^2 \right) \right)
\]
\[
= \sum_{p=1}^{n} \frac{\zeta_p^2}{\zeta_p} \left( \sum_{j=1}^{n} \left( \frac{\zeta_j^2}{\zeta_j} \left| f'_j(\phi_j^k) - f'_j(\tilde{\alpha}^*) \right|^2 \right) \right)
\]
\[
\leq \max_j \left( 1 - \tilde{\zeta}_j \right) \sum_{j=1}^{n} \frac{\zeta_j^2}{\zeta_j} \left| f'_j(\phi_j^k) - f'_j(\tilde{\alpha}^*) \right|^2 + \sum_{p=1}^{n} \frac{\zeta_p^2}{\zeta_p} \left| f'_p(u_k) - f'_p(\tilde{\alpha}^*) \right|^2
\]
\[
\leq \max_j \left( 1 - \tilde{\zeta}_j \right) Q_k + S_n L^2 \left| u_k - \tilde{\alpha}^* \right|^2
\]

\[\square\]

**Lemma 5.** Let \( P_k = (f'_{i_k}(u_k) - f'_{i_k}(\phi_{i_k}^k)) \frac{\zeta_{i_k}}{\zeta_{i_k}} + \sum_{j=1}^{n} f'_j(\phi_j^k)\zeta_j \) and \( T_k = P_k - \nabla \tilde{J}(\tilde{\alpha}^*) \), then we have the following properties:

(37) \[ E[P_k|F_k] = \nabla \tilde{J}(u_k) \]

(38) \[ E[T_k|F_k] = \nabla \tilde{J}(u_k) - \nabla \tilde{J}(\tilde{\alpha}^*) \]

(39) \[ E[\|T_k\|^2|F_k] \leq 9\tilde{S}_n L^2 \| u_k - \tilde{\alpha}^* \|^2 + 8Q_k \]

where \( \tilde{S}_n \) is defined as in (30).

**Proof.** Again, we further condition on the possible values taken by the random variable \( i_k \), thus obtaining:

\[
E[P_k|F_k] = E \left[ \left( (f'_{i_k}(u_k) - f'_{i_k}(\phi_{i_k}^k)) \frac{\zeta_{i_k}}{\zeta_{i_k}} + \sum_{j=1}^{n} f'_j(\phi_j^k)\zeta_j \right) | F_k \right]
\]
\[
= \sum_{p=1}^{n} E \left[ \left( (f'_{i_k}(u_k) - f'_{i_k}(\phi_{i_k}^k)) \frac{\zeta_{i_k}}{\zeta_{i_k}} + \sum_{j=1}^{n} f'_j(\phi_j^k)\zeta_j \right) | F_k, i_k = p \right] \tilde{\zeta}_p
\]
\[
= \sum_{j=1}^{n} f'_j(u_k) \frac{\zeta_j}{\zeta_j} - \sum_{j=1}^{n} f'_j(\phi_j^k) \frac{\zeta_j}{\zeta_j} + \sum_{j=1}^{n} f'_j(\phi_j^k)\zeta_j
\]
\[
= \sum_{j=1}^{n} f'_j(u_k)\zeta_j = \nabla \tilde{J}(u_k)
\]

which proves (37). We see from this that \( P_k \) is an unbiased estimator of \( \nabla \tilde{J}(u_k) \), when conditioned to \( F_k \), which represents the main difference with SAG, and simplifies the convergence proof. Equation (38) follows straightforwardly:

\[ E[T_k|F_k] = \nabla \tilde{J}(u_k) - \nabla \tilde{J}(\tilde{\alpha}^*). \]
We prove now (39).
\[
E[||T_k||^2 |F_k|] = E[||T_k - E[T_k |F_k]|^2 |F_k] + E[|E[T_k |F_k]|^2]
= E[||P_k - \nabla \tilde{J}(u_k)||^2 |F_k] + ||\nabla \tilde{J}(u_k) - \nabla \tilde{J}(\hat{u})||^2
= E[\left(\sum_{j=1}^n (f''_j(u_k) - f''_j(\phi^k_j)) \frac{\zeta_{ik}}{\zeta_{ik}} \right) + \sum_{j=1}^n (f_j'(\phi^k_j)) \zeta_j - \sum_{j=1}^n (f_j'(u_k)) \zeta_j ||^2 |F_k] + ||\nabla \tilde{J}(u_k) - \nabla \tilde{J}(\hat{u})||^2
\leq 2E[\left(\sum_{j=1}^n (f''_j(u_k) - f''_j(\phi^k_j)) \frac{\zeta_{ik}}{\zeta_{ik}} \right) + 2E[\left(\sum_{j=1}^n (f_j'(\phi^k_j)) ||^2 |F_k] + ||\nabla \tilde{J}(u_k) - \nabla \tilde{J}(\hat{u})||^2
\]

The first part A can be split as
\[
A = E[\left(\sum_{j=1}^n (f''_j(u_k) - f''_j(\phi^k_j)) \frac{\zeta_{ik}}{\zeta_{ik}} \right) + \sum_{j=1}^n (f_j'(\phi^k_j)) \zeta_j - \sum_{j=1}^n (f_j'(u_k)) \zeta_j ||^2 |F_k]
\leq 2E[\left(\sum_{j=1}^n (f''_j(u_k) - f''_j(\phi^k_j)) \frac{\zeta_{ik}}{\zeta_{ik}} \right) + 2E[\left(\sum_{j=1}^n (f_j'(\phi^k_j)) ||^2 |F_k]
\]

with
\[
T_1 \leq L^2 ||u_k - \hat{u}||^2 \frac{\zeta_{ik}}{\zeta_{ik}} = L^2 ||u_k - \hat{u}||^2 \tilde{S}_n
\]

The term T_2 can be developed as a sum over the possible values of i_k:
\[
T_2 = E[\sum_{j=1}^n \left(\frac{\zeta_{ik}}{\zeta_{ik}} \right) ||f_j'(\phi^k_j) - f_j'(u_k)||^2 |F_k] = \sum_{j=1}^n \left(\frac{\zeta_{ik}}{\zeta_{ik}} \right) ||f_j'(\phi^k_j) - f_j'(\hat{u})||^2
\]

Moreover
\[
B = E[\sum_{j=1}^n \zeta_j (f_j'(\phi^k_j) - f_j'(u_k)) ||^2 |F_k] \leq \left(\sum_{j=1}^n \zeta_j ||f_j'(\phi^k_j) - f_j'(u_k)|| \sqrt{\frac{\zeta_j}{\zeta_j}} \right)^2 \leq \left(\sum_{j=1}^n \left(\frac{\zeta_{ik}}{\zeta_{ik}} \right) \right) \left(\sum_{j=1}^n \zeta_j \right)^2 \leq 2 \sum_{j=1}^n \left(\frac{\zeta_{ik}}{\zeta_j} \right) ||f_j'(\phi^k_j) - f_j'(\hat{u})||^2 + 2 \sum_{j=1}^n \left(\frac{\zeta_{ik}}{\zeta_j} \right) ||f_j'(u_k)||^2 \leq 2Q_k + 2L^2 \tilde{S}_n ||u_k - \hat{u}||^2
\]

Finally
\[
C = \left(\sum_{j=1}^n \zeta_j (f_j'(u_k) - f_j'(\hat{u})) \right)^2 \leq \left(\sum_{j=1}^n \zeta_j \left||f_j'(u_k) - f_j'(\hat{u})\right|| \right)^2 \leq L^2 ||u_k - \hat{u}||^2 \left(\sum_{j=1}^n \zeta_j \right)^2 \leq \tilde{S}_n L^2 \tilde{S}_n ||u_k - \hat{u}||^2
\]

which completes the proof.
Lemma 6. Let $\alpha > 0$ and let Assumptions 1 and 2 hold. If $u_k$ denotes the $k$-th iterate of SAGA Algorithm 2 and $\tilde{u}^*$ is the solution of the OCP [15], then there exist $D_1, D_2 \in \mathbb{R}_+$ such that:

$$
\mathbb{E}[\|u_{k+1} - \tilde{u}^*\|^2 + \alpha Q_{k+1}|F_k] \leq D_1\|u_k - \tilde{u}^*\|^2 + D_2\alpha Q_k
$$

with $D_1 = 1 - l\tau + (\alpha S_n + 8\tau^2 S_\tau)L^2 + \tau^2 L^2$, $D_2 = 1 - \zeta_{\min} + \frac{8\tau^2}{\alpha}$, $\zeta_{\min} = \min_j \zeta_j$ and $S_n$ as in Lemma 4.

Proof. Using that $\nabla \tilde{J}(\tilde{u}^*) = 0$, we have

$$
\|u_{k+1} - \tilde{u}^*\|^2 = \|u_k - \tilde{u}^* - \tau (P_k - \nabla \tilde{J}(\tilde{u}^*))\|^2 = \|u_k - \tilde{u}^*\|^2 - 2\tau \langle u_k - \tilde{u}^*, T_k \rangle + \tau^2 \|T_k\|^2.
$$

Let us develop now $\|u_{k+1} - \tilde{u}^*\|^2 + \alpha Q_{k+1}$, using Lemmas 4 and 5.

$$
\mathbb{E}[\|u_{k+1} - \tilde{u}^*\|^2 + \alpha Q_{k+1}|F_k] = \mathbb{E}[\|u_k - \tilde{u}^*\|^2|F_k] - 2\tau \mathbb{E}[\langle u_k - \tilde{u}^*, T_k \rangle|F_k] + \tau^2 \mathbb{E}[\|T_k\|^2|F_k] + \alpha \mathbb{E}[Q_{k+1}|F_k]
$$

$$
= \|u_k - \tilde{u}^*\|^2 - 2\tau \langle u_k - \tilde{u}^*, \nabla \tilde{J}(\tilde{u}^*) \rangle + \tau^2 \|T_k\|^2 + \alpha \mathbb{E}[Q_{k+1}|F_k]
$$

$$
\leq \|u_k - \tilde{u}^*\|^2 - l\tau \|u_k - \tilde{u}^*\|^2 + \tau^2 \left(9 S_n L^2\|u_k - \tilde{u}^*\|^2 + 8Q_k\right)
$$

$$
+ \alpha \left(\max_j (1 - \zeta_j) Q_j + S_n L^2\|u_k - \tilde{u}^*\|^2\right)
$$

$$
\leq \left(1 - l\tau + (\alpha S_n + 9\tau^2 S_\tau)L^2\right)\|u_k - \tilde{u}^*\|^2 + \left(\max_j (1 - \zeta_j) + 8\frac{\tau^2}{\alpha}\right)\alpha Q_k
$$

We are now ready to state the final convergence result. For this, we need to find the right choice of $\alpha > 0$ and $\tau$ s.t.

\begin{align}
\text{(40)} & \quad 1 - l\tau + (\alpha S_n + 9\tau^2 S_\tau)L^2 = D_1 < 1 \\
\text{(41)} & \quad \max_j (1 - \zeta_j) + 8\frac{\tau^2}{\alpha} = D_2 < 1
\end{align}

One particular choice that guarantees an exponential in $k$ convergence rate is shown in the following Theorem.

Theorem 7. Let Assumptions 1, 2 and 5 hold and let us define:

$$
\tilde{\zeta}_j = \frac{1}{n}, \quad N = 25 S_\tau, \quad \tau = \frac{1}{2NL^2}, \quad \alpha = 16\sigma^2.
$$

with $S_\tau$ as in Assumption 4. Then, we have

$$
\mathbb{E}[\|u_{k+1} - \tilde{u}^*\|^2 + \alpha Q_{k+1}] \leq \rho \mathbb{E}[\|u_k - \tilde{u}^*\|^2 + \alpha Q_k]
$$

with $\rho = \min\{1 - \frac{\tau^2}{4NL^2}, 1 - \frac{1}{2n}\} \in (0, 1)$. Notice, in particular, that $N$ and $\tau$ do not depend on $n$.

Proof. The particular choice of $\{\zeta_j\} = \{\tilde{\zeta}_j\}$, $\alpha$, and $\tau$ implies $D_1 \leq 1 - \frac{\tau^2}{4NL^2}$, $D_2 = 1 - \frac{1}{2n}$, where we have exploited the fact that $n S_n = S_\tau$ for $\tilde{\zeta}_j = 1/n$. Hence,

$$
\mathbb{E}[\|u_{k+1} - \tilde{u}^*\|^2 + \alpha Q_{k+1}|F_k] \leq D_1\|u_k - \tilde{u}^*\|^2 + D_2\alpha Q_k \leq \max(D_1, D_2)\|u_k - \tilde{u}^*\|^2 + \alpha Q_k
$$

The final result is obtained by taking a further expectation over $(i_0, \ldots, i_{k-1})$.

Corollary 2. Under Assumptions 1, 2 and 5 if $u_k$ denotes the $k$-th iterate of SAGA described in Algorithm 2 and $\tilde{u}^*$ is the solution of the semi-discrete OCP [15], then there exists $D_3 > 0$ such that:

\begin{align}
\text{(42)} & \quad \mathbb{E}\left[\|u_k - \tilde{u}^*\|^2\right] \leq D_3(1 - \epsilon)^k
\end{align}

with $\epsilon = \min\left\{\frac{\tau^2}{4NL^2}, \frac{1}{2n}\right\}$. Proof. This result is a direct application of Theorem 7.

Remark 2. Theorem 7 generalized for any $\tau \in (0, \frac{1}{4NL^2})$, in which case $D_3(\tau) = 1 - l\tau + \tau^2 L^2 N (1 - \frac{\tau^2}{4NL^2}, 1)$.

We finish this subsection by showing that Assumption 5 holds, in the case of Gauss-Legendre quadrature.

Lemma 7. In the setting of uniform random variables $\xi$ and tensorized Gauss-Legendre quadrature formulas, choosing $\zeta_0 = \frac{1}{n}$, then Assumption 5 holds.
Proof. As shown in [13, page 353, (15.3.10)], the weights of the Gauss-Legendre quadrature formula satisfy

\[ \zeta_p \lesssim \frac{1}{n}. \]

Hence, for a tensor quadrature with \((n_1, \ldots, n_M)\) points in each variables, and a multi-index \(p = (p_1, \ldots, p_M)\), with \(1 \leq p_i \leq n_i\) as \( n = \prod_i n_i \) we have

\[ \zeta_p = \prod_{i=1}^{M} \zeta_{p_i} \lesssim \prod_{i=1}^{M} \frac{1}{n_i} \]

and

\[
\sum_p \frac{\zeta_p^2}{\zeta_p} = \sum_p \zeta_p^2 \lesssim \sum_{p_1=1}^{n_1} \cdots \sum_{p_M=1}^{n_M} \left( \prod_{i=1}^{M} \frac{1}{n_i} \right) \prod_{i=1}^{M} n_i \\
= \sum_{p_1=1}^{n_1} \cdots \sum_{p_M=1}^{n_M} \prod_{i=1}^{M} \frac{n_i}{n_i} \\
= \prod_{i=1}^{M} \frac{1}{n_i} \prod_{i=1}^{M} n_i \\
= 1
\]

with constant in the symbol \( \lesssim \) independent of \((n_1, \ldots, n_M)\), but depending exponentially on \(M\).

\[ \square \]

Remark 3. Result of Lemma 4 still holds for Gauss-Jacobi abscissas, as proven in [13, page 353, (15.3.10)].

3.3. Complexity analysis of SAGA. The convergence result stated in Theorem 3 for the semi-discrete OCP [15] applies equally well to the discrete OCP [18] with the same constants (thanks to the fact that the FE approximation functions \( f_i^h(u^h) \) satisfy strongly convexity and Lipschitzianity inequalities as in Lemmas 1 and 2 with the same constants). We now analyze the complexity of the SAGA algorithm 2 in terms of computational work \(W\) versus accuracy \( \text{tol} \). The complexity analysis is based on the following error splitting into FE discretization error, Gauss-Legendre quadrature error and SAGA optimization error when stopping the SAGA algorithm at iteration \(k\) leading to the following result.

Theorem 8. With same notations of Corollary 2, if \( \hat{u}^{h_{SAGA_k}} \) denotes the approximated optimal control computed by using successively a FE approximation, full tensor Gauss-Legendre quadrature formula and SAGA method, then the MSE is bounded by:

\[
\mathbb{E}[\|\hat{u}^{h_{SAGA_k}} - \hat{u}^*\|^2] \leq C_1(1 + \epsilon)^k + C_2 \sum_{n=1}^{M} e^{-\gamma n q_n} + C_3 h^{2r+2}
\]

with \( \epsilon = \frac{1}{2n} = \frac{\prod_{i=1}^{M} n_i^{-1}}{2}, \) assuming that \( n > \frac{50SL^2}{\epsilon} \), and with constants \( C_1, C_2 \) and \( C_3 \) independent of \(k, \{q_n\}_n\) and \(h\).

Proof. We can decompose the total error using the three successive approximations presented in Section 1 and 3 FE discretization, quadrature formula and SAGA optimization procedure:

\[
\mathbb{E}[\|\hat{u}^{h_{SAGA_k}} - \hat{u}^*\|^2] \leq 3 \mathbb{E}[\|\hat{u}^{h_{SAGA_k}} - \hat{u}^{h*}\|^2] + 3 \mathbb{E}[\|\hat{u}^{h*} - u^{h*}\|^2] + 3 \|u^{h*} - \hat{u}^*\|^2
\]

where \( \hat{u}^{h*} \) is the optimal solution of the fully-discrete OCP [18] and \( u^{h*} \) is the optimal control of the FE discretized OCP [12]. The result is straightforward using the bounds in Corollary 2, Theorem 2 and Theorem 3.

\[ \square \]

Corollary 3. In order to achieve a given tolerance \( O(\text{tol}) \), i.e. to guarantee that \( \mathbb{E}[\|\hat{u}^{h_{SAGA_k}} - \hat{u}^*\|^2] \lesssim \text{tol}^2 \), the total required computational work is bounded by

\[
W \lesssim (\log(\text{tol}^{-1}))^{M+1} \text{tol}^{\frac{d}{\gamma+1}}.
\]

where we assume that the primal and dual problems can be solved, using a triangulation with mesh size \(h\), in computational time \( C_h = O(h^{-\gamma}) \). Here, \( \gamma \in [1, 3] \) is a parameter representing the efficiency of the linear solver used (e.g. \( \gamma = 3 \) for a direct solver and \( \gamma = 1 \) up to a logarithm factor for an optimal multigrid solver), while \(d\)
is the dimension of the physical space $D \subset \mathbb{R}^d$. The memory space required to store the history of the computed gradients scales as

$$\text{storage} = O \left( \left( \log(\text{tol}^{-1}) \right)^M \text{tol}^{-\frac{d}{2+}} \right)$$

Proof. Using Theorem 5 as we want to guarantee an error of order $O(\text{tol})$, we can equalize the three terms on the right hand side of (43) to tol$^2$ and finally get:

$$h = O(\text{tol}^{-\frac{1}{M+1}}), \quad q_j = \frac{2}{s_j} \log(\text{tol}^{-1}), \quad n = \left( \frac{2}{R} \log(\text{tol}^{-1}) \right)^M, \quad k = \frac{2 \log(\text{tol}^{-1})}{-\log(1 - \frac{1}{2s})}$$

with $R = \left( \prod_j s_j \right)^{1/M}$. So we obtain asymptotically

$$k \sim 4n \log(\text{tol}^{-1}) \sim 4 \left( \frac{2}{R} \log(\text{tol}^{-1}) \right)^M \log(\text{tol}^{-1}) = O \left( \left( \log(\text{tol}^{-1}) \right)^{M+1} \right)$$

If $W$ denotes the computational work (proportional to time if we do not use any parallel computing strategy), we have

$$W = O \left( \left( \log(\text{tol}^{-1}) \right)^{M+1} \text{tol}^{-\frac{d}{2+}} \right).$$

The memory space required to store the history of all the $n$ computed gradients is proportional to $nh^{-d}$, so:

$$\text{storage} = O \left( \left( \log(\text{tol}^{-1}) \right)^M \text{tol}^{-\frac{d}{2+}} \right).$$

\[ \square \]

The computational work and storage requirements for SAGA stated in Corollary 3 are reported in Table 2. For comparison, we state in the same Table also the complexity and storage requirement of the standard Stochastic Gradient algorithm, as well as the Full Gradient algorithm, both based on the same quadrature formula and FE approximation as for SAGA (we refer to [10] where these results have been derived in the context of a Monte Carlo approximation). A naive implementation of the FG algorithm would require to store the gradient computed in each quadrature point, hence a storage of $O(\text{tol}^{-d})$ . Alternatively, one can store only the partial weighted sum of the gradients and update it as soon as the gradient in a new quadrature point has been computed, which brings down the storage to $O(h^{-d})$.

4. Numerical results

In this section we verify the assertions on the order of convergence and computational complexity stated in Theorem 5 and Corollary 3. For this purpose, we consider the optimal control problem (9) in the domain $D = (0, 1)^2$ with $g = 1$ and the following random diffusion coefficient:

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

with $(x_1, x_2) \in D$, $\text{var} = \exp(-1.125)$ and $\xi = (\xi_1, \ldots, \xi_4)$ with $\xi_i \overset{\text{id}}{\sim} U([-1, 1])$ (this test case is taken from [9]). We have chosen $\beta = 10^{-4}$ as the price of energy (regularization parameter) in the objective functional. For the FE approximation, we have considered a structured triangular grid of mesh size $h$ where each side of the domain $D$ is divided into $1/h$ sub-intervals and used piece-wise linear finite elements (i.e. $r = 1$). For the approximation of the expectation in the objective functional, we have used a full tensor Gauss-Legendre quadrature formula with the same number $q$ of quadrature knots in each random variable $\xi_j$, $j = 1, \ldots, 4$. All calculations have been performed using the FE library Freefem++ [8].

|         | FG          | SG          | SAGA        |
|---------|-------------|-------------|-------------|
| storage | $\text{tol}^{-\frac{d}{2+}}$ | $\text{tol}^{-\frac{d}{2+}}$ | $\left( \log(\text{tol}^{-1}) \right)^M \text{tol}^{-\frac{d}{2+}}$ |

Table 2. Computational work and required storage memory for the modified Algorithm 1 and 2 to solve the OCP.
4.1. **Performance of SAGA and comparison with FG.** In this subsection, we consider the SAGA method using a fixed mesh size $h = 2^{-3}$ and study its convergence for different levels of the full tensor Gauss-Legendre quadrature formula, i.e. a different number $q$ of points in each random variable (the total number of quadrature points being $q^3$). For each $q$, we compute a reference solution using 50000 SAGA iterations (using again the same FE mesh size $h = 2^{-3}$). Then, we perform 5000 SAGA iteration and compare the error $u_k - u_{ref}$ w.r.t. the reference solution. We repeat the computation 20 times, independent, to estimate the log-mean error log($E[||error||]$) (hereafter log($\cdot$) refers to the base 10 logarithm). In all cases we have used a step-size $\tau = 1000$. We show in Figure 1 the convergence plots of log($E[||error||]$) versus $k$, for $q \in \{1, \ldots, 10\}$ and in Figure 2 a zoom on the first 500 iterations. We can observe two regimes: a first one over the first 20 iterations, of faster exponential convergence, and a second one afterwards, of slower, but still exponential convergence. Then, in both cases is plotted against the number of PDE solves. The plot shows a fastest convergence of FG than SAGA, asymptotically. However, SAGA features a smaller error in the pre-asymptotic regime, and delivers an

![Figure 1: Convergence of SAGA for different $q$ and fixed FE mesh (reference solution computed with the same FE mesh and quadrature with $q^3$ nodes)](image-url)

![Figure 2: Convergence plots for different $q$ and fixed step-size](image-url)

where $\hat{E}$ is the Gauss-Legendre quadrature formula defined in (14), and $y_k(\cdot)$ is the solution of (3). The error in both cases is plotted against the number of PDE solves. The plot shows a fastest convergence of FG than SAGA, asymptotically. However, SAGA features a smaller error in the pre-asymptotic regime, and delivers an
acceptable solution, from a practical point of view, already before two full iterations of FG (*MM: 2500 PDE solves). This makes it attractive in a limited budget context.

4.2. Complexity results for the SAGA algorithm. We investigate here the convergence of the method defined in Algorithm 2, for which we recall the error bound (43) in the case of piece-wise linear FE (i.e. $r = 1$) and a 4-dimensional stochastic variable (i.e. $M = 4$):

$$
\mathbb{E}[\|\hat{u}_{\text{SAGA}_k} - u^*\|^2] \leq C_1(1 - \epsilon(q))^k + C_2e^{-sq} + C_3h^4
$$

where $s$ is the rate of exponential convergence of the quadrature formula, and $q$ the number of knots used in each stochastic variable (isotropic case).

Estimation of the constants $C_1, \epsilon(q), C_2, s, C_3$. We have estimated the constants in (50) numerically.

- In order to estimate $C_1$ and $\epsilon$ in (50), we used a fixed mesh with $h = 2^{-3}$ and a fixed quadrature formula for both reference solution and SAGA iterations. By doing so, we remove the two last terms in the bound (50), and only keep the first one of which we want to estimate the constants $C_1$ and $\epsilon(q)$. Using Figures 1 and 3 previously described, we estimate $C_1 \in [1200, 4500]$ and $\epsilon(q) \approx 0.2 q^4$.

- To estimate the constant $C_2$ and $s$ in (50), we use a mesh of size $h = 2^{-4}$ for both the reference solution and the optimal control with quadrature. First, we computed the reference solution for a fine Gauss-Legendre quadrature formula, i.e. $q = 8$, using the FG algorithm up to iteration 300. Then we computed the error for the approximated optimal control using only $q \in \{1, \ldots, 5\}$ points in each random variable, using again the FG algorithm up to iteration 100. Results are detailed in Table 3 and plotted in Figure 6. The error is the difference between the estimated optimal control using $q \in \{1, \ldots, 5\}$ knots in the quadrature formula, and the optimal control computed for $q = 8$. We estimate $C_2 \approx 57.4$ and $s \approx 5.89$.

- Finally, to estimate the third term $C_3$, we used the FG algorithm up to iteration 300, with a step-size $\tau = 2000$, on a quadrature formula with $q = 1$ knot in each random variable, and computed the reference solution on a fine mesh with $h = 2^{-7}$. Then we run FG with the same step size $\tau = 2000$ and quadrature ($q = 1$) on coarser meshes with $h = 2^{-1}, \ldots, 2^{-6}$. Results are shown in Table 4. The Table confirms a convergence $O(h^4)$ of the squared error with an estimated constant $C_3 \approx 1170$. To double check our estimate, we repeated the estimation using the SAGA algorithm: for the reference solution,
Figure 3. Assessment of SAGA convergence rate $C_1(1-\epsilon)^k$ in (50): Top: estimated rate $\epsilon_{est}$ over theoretical rate $\epsilon_{th} = \frac{1}{2\theta}$ versus $q$. Bottom: estimated constant $C_1$ versus $q$.

Table 3. Quadrature error on the optimal control, versus the number of knots $q$ used in each random variable.

| $q$ | $\|error\|_{\mathbf{h}}$ |
|-----|-------------------|
| 1   | 1.22e-1           |
| 2   | 5.27e-4           |
| 3   | 1.45e-6           |
| 4   | 3.35e-9           |
| 5   | 7.53e-12          |

Table 4. FE discretization error on the optimal control, using FG, or SAGA, versus the characteristic mesh size $h^{-1}$.

| $h^{-1}$ | $error$ (FG, $q = 1$) | $error$ (SAGA, $q = 1$) |
|----------|------------------------|-------------------------|
| 2        | 7.83                   | 7.82                    |
| 4        | 2.19                   | 2.19                    |
| 8        | 5.46E-01               | 5.48E-01                |
| 16       | 1.36E-01               | 1.37E-01                |
| 32       | 3.34E-02               | 3.42E-02                |
| 64       | 8.34E-03               | 8.47E-03                |
| 128      | 2.09E-03               | 2.04E-03                |

we used a mesh size $h = 2^{-9}$, and $q = 2$ points in the quadrature formula, and SAGA algorithm up to iteration 20000. Then we computed 10 repetitions of SAGA using mesh sizes $h = 2^{-1}, \ldots, 2^{-7}$, up to iteration 10000 and computed the average error. Results are shown in the third column of Table 4. The
PDE-CONSTRAINED OPTIMAL CONTROL PROBLEMS WITH UNCERTAIN PARAMETERS USING SAGA

Figure 4. SAGA sensitivity to $\tau$ for $q = 5$.

Table 5. Final error over 20 i.i.d. realizations of SAGA, versus the target tolerance $tol$.

| $tol$   | $1/h$ | $q$ | $n_{tot}$ | $#\text{knts}$ | $k_{max}$ | comp. cost | storage | avg $|u_{ref} - u_{h,q}^{k_{max}}|$ |
|---------|-------|-----|-----------|----------------|-----------|------------|---------|-------------------------------|
| 3.16E-01| 11    | 1   | 1         | 46             | 1.11E+04  | 1.21E+02  | 2.59E-01|
| 1.00E-01| 19    | 2   | 16        | 243            | 1.75E+05  | 5.78E+03  | 8.02E-02|
| 3.16E-02| 33    | 2   | 16        | 339            | 7.38E+05  | 1.74E+04  | 2.68E-02|
| 1.00E-02| 59    | 3   | 81        | 1896           | 1.32E+07  | 2.82E+05  | 8.29E-03|
| 3.16E-03| 105   | 3   | 81        | 2417           | 5.33E+07  | 8.93E+05  | 2.62E-03|
| 1.00E-03| 185   | 4   | 256       | 9298           | 6.36E+08  | 8.76E+06  | 8.00E-04|
| 3.16E-04| 329   | 4   | 256       | 18947          | 4.74E+09  | 2.77E+07  | 2.80E-04|

The main limitation we see for this method is the required memory space, since the storage increases as the desired tolerance gets smaller and will reach at some point the memory limit of the employed machine.
5. Conclusions (Fabio)

In this work, we have proposed a SAGA algorithm to solve numerically a quadratic risk-averse optimal control problem for an elliptic PDE with random coefficients, where the expectation in the objective functional has been approximated by a Gauss-Legendre quadrature formula whereas the elliptic PDE has been discretized by finite elements. The SAGA algorithm is a Stochastic Gradient type algorithm with a fixed-length memory term, which computes at each iteration the gradient of the objective functional in only one quadrature point, randomly chosen from a possibly non-uniform distribution. We have shown that the asymptotically optimal sampling distribution is indeed the uniform one, over the quadrature points. We have also shown that, when equilibrating the three sources of errors, namely the finite element discretization error, the quadrature error and the error due to the SAGA optimization algorithm, the overall complexity, in terms of computational work versus prescribed tolerance, is asymptotically the same as the one of a full gradient method (i.e. a gradient method that sweeps over all quadrature points at each iteration), as the tolerance goes to zero. However, as illustrated by our numerical experiments, the advantage of SAGA with respect to FG is in the pre-asymptotic regime, as acceptable solutions may be obtained already before a full sweep over all quadrature points.

The full tensor Gauss-Legendre quadrature formula considered in this work is affected by the curse of dimensionality, hence applicable only to problems for which the randomness can be described in terms of a small number of random variables. To overcome such curse of dimensionality, one could use sparse quadratures instead \[1, 2, 9\], whose weights, however, are not all positive. The result in Lemma 6 is still valid, as long as the approximate functional \( \hat{J} \) satisfies a strong convexity condition,

\[
\frac{l}{2} \| u_1 - u_2 \|^2 \leq \langle \nabla_u \hat{J}(u_1) - \nabla_u \hat{J}(u_2), u_1 - u_2 \rangle \quad \forall u_1, u_2 \in U,
\]

which might not be guaranteed for a given number of quadrature points. Also, because of the presence of negative weights, the quantity \( \tilde{S}_n \) might not be uniformly bounded in \( n \), therefore, the results in Theorem 7 and Corollary 2 might not apply to this case. These issues will be further investigated in a future work.
Figure 6. Fitting the quadrature (squared) error model $C_2e^{-sq}$ in [50].

Acknowledgments

F. Nobile and M. Martin have received support from the Center for ADvanced MOdeling Science (CADMOS).

References

[1] I. Babuška, F. Nobile, and R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM review 52 (2010), no. 2, 317–355.
[2] A. Borzi, Multigrid and sparse-grid schemes for elliptic control problems with random coefficients, Computing and Visualization in Science 13 (2010), no. 4, 153–160.
[3] H-J. Bungartz and M. Griebel, Sparse grids, Acta Numer. 13 (2004), 147–269. MR 2249147
[4] A. Defazio, F. Bach, and S. Lacoste-Julien, Saga: A fast incremental gradient method with support for non-strongly convex composite objectives, Advances in Neural Information Processing Systems 27 (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), Curran Associates, Inc., 2014, pp. 1646–1654.
[5] A. Dieuleveut and F. Bach, Nonparametric stochastic approximation with large step-sizes, The Annals of Statistics 44 (2016), no. 4, 1363–1399.
[6] O. G. Ernst, B. Sprungk, and L. Tamellini, Convergence of sparse collocation for functions of countably many Gaussian random variables (with application to elliptic PDEs), SIAM J. Numer. Anal. 56 (2018), no. 2, 877–905. MR 3784113
[7] A-L. Haji-Ali, F. Nobile, L. Tamellini, and R. Tempone, Multi-index stochastic collocation convergence rates for random PDEs with parametric regularity, Found. Comp. Math. 16 (2016), no. 6, 1555–1605.
[8] F. Hecht, New development in freefem++, J. Numer. Math. 20 (2012), no. 3-4, 251–265. MR 3043640
[9] H.-C. Lee and M. D. Gunzburger, Comparison of approaches for random pde optimization problems based on different matching functionals, Computers and Mathematics with Applications 73 (2017), no. 8, 1657 – 1672.
[10] M. C. Martin, S. Krumscheid, and F. Nobile, Analysis of stochastic gradient methods for pde-constrained optimal control problems with uncertain parameters, (2018), 39.
Figure 7. Computational cost (model) versus estimated average error for the SAGA Algorithm.

[11] F. Nobile, L. Tamellini, and R. Tempone, Convergence of quasi-optimal sparse grid approximation of Hilbert-space-valued functions: application to random elliptic PDEs, Numer. Math. 134 (2016), no. 2, 343–388.

[12] B.T. Polyak and A.B. Juditsky, Acceleration of stochastic approximation by averaging, SIAM Journal on Control and Optimization 30 (1992), no. 4, 838–855.

[13] H. Robbins and S. Monro, A stochastic approximation method, Ann. Math. Statist. 22 (1951), no. 3, 400–407.

[14] C. Schillings and C. Schwab, Sparse, adaptive Smolyak quadratures for Bayesian inverse problems, Inverse Problems 29 (2013), no. 6, 065011, 28. MR 3056084

[15] M. Schmidt, N. Le Roux, and F. Bach, Minimizing finite sums with the stochastic average gradient, Math. Program. 162 (2017), no. 1-2, Ser. A, 83–112. MR 3612933

[16] M. W. Schmidt, R. Babanezhad, M. O. Ahmed, A. Defazio, A. Clifton, and A. Sarkar, Non-uniform stochastic average gradient method for training conditional random fields, Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015.

[17] A. Shapiro, D. Dentcheva, and A. Ruszczynski, Lectures on stochastic programming, Society for Industrial and Applied Mathematics, 2009.

[18] Gabor Szeg, Orthogonal polynomials, vol. 23, American Mathematical Soc., 1939.

[19] J. Zech, D. Dung, and C. Schwab, Multilevel approximation of parametric and stochastic PDEs, Tech. Report SAM Report 2018-05, ETHZ, 2018.