Chapter 2
Open-Loop Control: The Stochastic Gradient Method

2.1 Introduction

The stochastic gradient method has a rather long history. The method foundations were given by Robbins and Monro [129] on the one hand, and by Kiefer and Wolfowitz [93] on the other. Later on, Polyak [120, 123] gave results about the convergence rate. Based on this work, Dodu et al. [57] studied the optimality of the stochastic gradient algorithm, that is, the asymptotic efficiency of the associated estimator. An important contribution by Polyak [121, 122] has been to combine stochastic gradient method and averaging techniques in order to reach the optimal efficiency.

Such methods have also been developed in the framework of Stochastic Approximation (SA) (see [98] for a review paper). The reference book by Kushner and Clark [96] presents the Ordinary Differential Equation method (ODE) in the nonconvex case, which makes it possible to perform a local convergence analysis for general stochastic algorithms. Other reference books are those of Duflo [59, 60] and again Kushner and Yin [97], including important topics as asymptotic normality or ways to deal with constraints. The reader is also referred to lecture notes by Delyon [54] giving a clear and detailed presentation of the subject.

The aim of this chapter is to detail the main methods available in order to analyze the behavior of stochastic gradient algorithms. After a brief discussion about open-loop optimization problems in Sect. 2.2, we present

- the general idea of stochastic gradient methods, the associated probabilistic framework, as well as "classical" theorems about almost-sure convergence (Robbins-Monro) and rate of convergence (Central Limit Theorem) in Sect. 2.3,
- a convergence result of the stochastic gradient algorithm in the framework of the Auxiliary Problem Principle in Sect. 2.4,
- the optimality analysis of the rate of convergence, that is, the optimal efficiency provided by the use of a matrix gain, and also by the averaging technique in Sect. 2.5,
- practical considerations about the numerical implementation of stochastic gradient algorithms in Sect. 2.6.

© Springer International Publishing Switzerland 2015

P. Carpentier et al., *Stochastic Multi-Stage Optimization*, Probability Theory and Stochastic Modelling 75, DOI 10.1007/978-3-319-18138-7_2

In this chapter, we often make use of several notions and terms specific to the optimization framework (proper function, lower semicontinuity, Lipschitz continuity, differentiability, gradient, strong convexity, strong monotonicity, coercivity, optimality conditions…). The reader is referred to Appendix A for the associated definitions and related properties.

2.2 Open-Loop Optimization Problems

We first discuss the notion of *open-loop optimization*, that is, the situation in which the decision maker is only aware of the a priori probability distribution of the random variables involved in the problem as mentioned in Sect. 1.2.2.

2.2.1 Problem Statement

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and let $\boldsymbol{W}$ be a random variable defined on Ω and taking its values in a measurable space $(\mathbb{W}, \mathcal{W})$. The probability distribution $\mathbb{P} \circ \boldsymbol{W}^{-1}$ of $\boldsymbol{W}$ is denoted by μ. Let $\mathbb{U}$ be a Hilbert space (with scalar product $\langle \cdot\,, \cdot \rangle$ and norm $\|\cdot\|$), and let U^{ad} be a non empty closed convex subset of $\mathbb{U}$. We consider a real-valued measurable function j defined on $\mathbb{U} \times \mathbb{W}$. We denote by $J(u)$ the expectation of the random variable $j(u, \boldsymbol{W})$ (we assume that the expectation exists for all $u \in U^{\mathrm{ad}}$):

$$J(u) = \mathbb{E}\big(j(u, \boldsymbol{W})\big) = \int_{\Omega} j\big(u, \boldsymbol{W}(\omega)\big)\, \mathrm{d}\mathbb{P}(\omega) = \int_{\mathbb{W}} j(u, w)\, \mathrm{d}\mu(w).$$

We assume that j is differentiable w.r.t. u, and that conditions for differentiating under the integral sign hold true. This classical issue is addressed by Integration Theory and can be found in [137, Sect. 3, Théorème 6.3.5] (see also [134] for a similar result about subdifferentiation). Then J is differentiable, its gradient is denoted by $\nabla J(u)$ and we have that

$$\nabla J(u) = \mathbb{E}\big(\nabla_u j(u, \boldsymbol{W})\big), \tag{2.1}$$

where $\nabla_u j$ is the gradient of j w.r.t. u. We are interested in the following optimization problem:

$$\min_{u \in U^{\mathrm{ad}}} J(u). \tag{2.2}$$

We consider here *open-loop* optimization problems, that is, problems in which the decision variable u is chosen without further information about $\boldsymbol{W}$ than its probability distribution.

Under standard convexity and differentiability assumptions, provided that we are able to compute the gradient of J for each $u \in U^{\mathrm{ad}}$, we may use a gradient-like algorithm (such as steepest descent, conjugate gradient, quasi-Newton, etc.) in order to compute the solution of Problem (2.2). The simplest is the projected gradient algorithm which reads

$$u^{(k+1)} = \mathrm{proj}_{U^{\mathrm{ad}}}\left(u^{(k)} - \epsilon \nabla J(u^{(k)})\right),$$

where ϵ is the gradient step size. Actually, this algorithm directly tackles the *deterministic* optimization problem (2.2) whereas the stochastic aspect is fully handled by the computation of the expectation involved in the expression (2.1) of $\nabla J(u^{(k)})$. However, this operation may be exceedingly costly if not impossible when the dimension of the space $\mathbb{W}$ is large.

Consider Problem (2.2), and replace $J(u)$ by its expression:

$$\min_{u \in U^{\mathrm{ad}}} \mathbb{E}\big(j(u, \boldsymbol{W})\big). \tag{2.3}$$

A standard way to get around the difficulty of computing an expectation is to use the Monte Carlo approach (see Sect. B.7). Using this idea in our optimization framework leads to replace Problem (2.3) by the following approximation

$$\min_{u \in U^{\mathrm{ad}}} \frac{1}{k} \sum_{l=1}^{k} j(u, w^l), \tag{2.4}$$

where $(w^1, \ldots, w^k)$ is a realization of a k-sample of $\boldsymbol{W}$.[1] Note that the gradient of the cost function of Problem (2.4), namely

$$\frac{1}{k} \sum_{l=1}^{k} \nabla_u j(u, w^l),$$

corresponds to a Monte Carlo approximation of the "true" gradient $\nabla J(u)$. This approach is known as the *Sample Average Approximation* (SAA), which is briefly presented in Sect. 2.5.3 (see [141, Chap. 5] for a detailed presentation). A drawback of the formulation (2.4) is that the sample size k is fixed prior to the resolution: one needs to solve a new optimization problem when enriching the initial sample with new realizations.

The stochastic gradient method aims to overcome the two difficulties mentioned above (that is, computing the true expectation or choosing the size of the sample prior to the resolution). In the manner of Sample Average Approximation, it uses an "easily computable" approximation of the gradient ∇J based on a sampling of $\boldsymbol{W}$.

[1] Recall that a k-sample of $\boldsymbol{W}$ is a sequence $(\boldsymbol{W}^1, \ldots, \boldsymbol{W}^k)$ of independent random variables with the same probability distribution as $\boldsymbol{W}$. See Sect. B.7.2 for further details.

Moreover, the samples are incorporated successively into the algorithm in order to produce a sequence of estimators converging towards the solution of Problem (2.3). In a sense, iterations of the gradient algorithm are also used to refine the Monte Carlo sampling process. Because of this sequential point of view in the introduction of new samples, the superscript l is now denoted (l) as an iteration index. The stochastic gradient method is presented in Sect. 2.3.

2.2.2 Sample Approximation in Stochastic Optimization

Suppose that we have built an approximation of Problem (2.3) using a k-tuple $(w^{(1)}, \dots, w^{(k)})$ of elements of $\mathbb{W}$ related to the random variable $\boldsymbol{W}$ (see (2.4) for an example). The solution $u^{(k)}$ of the approximated problem can be viewed as a (measurable) function of that sequence:

$$u^{(k)} = \varphi^{(k)}(w^{(1)}, \dots, w^{(k)}).$$

The performance $\mathbb{E}\big(j(\varphi^{(k)}(w^{(1)}, \dots, w^{(k)}), \boldsymbol{W})\big)$ of the approximated solution $u^{(k)}$ can also be viewed as a (measurable) function $\psi^{(k)}$ of the sequence $(w^{(1)}, \dots, w^{(k)})$. To alleviate the notation, we set:

$$J^{(k)} = \psi^{(k)}(w^{(1)}, \dots, w^{(k)}) = \mathbb{E}\big(j(\varphi^{(k)}(w^{(1)}, \dots, w^{(k)}), \boldsymbol{W})\big). \tag{2.5}$$

In the computation of $J^{(k)}$, it should be clear that the expectation operates on the random variable $\boldsymbol{W}$ whereas the $w^{(k)}$'s are considered as parameters (and therefore, the result of this calculation is also a function of those parameters). Suppose that those parameters are the result of random drawings: then $J^{(k)}$ is the realization of a random variable defined on another probability space that we are going to introduce now.

To be more specific about the approximation, suppose that the k-tuple $(w^{(1)}, \dots, w^{(k)})$ is a realization of a k-sample $(\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(k)})$ of $\boldsymbol{W}$. As explained in Sect. B.7.2, we have to deal with two different probability spaces: the random variable $\boldsymbol{W}$ is defined on the canonical probability space $(\Omega, \mathcal{A}, \mathbb{P})$ whereas the k-tuple $(\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(k)})$ is defined on $(\widetilde{\Omega}, \widetilde{\mathcal{A}}, \widetilde{\mathbb{P}})$, the infinite-dimensional product of the probability spaces $(\mathbb{W}, \mathcal{W}, \mu)$:

$$(\widetilde{\Omega}, \widetilde{\mathcal{A}}, \widetilde{\mathbb{P}}) = (\mathbb{W}^{\mathbb{N}}, \mathcal{W}^{\otimes\mathbb{N}}, \mu^{\otimes\mathbb{N}}).$$

Of course, $(\boldsymbol{W}, \boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(k)})$ can be identified with a $(k+1)$-sample, so that all random variables can be considered as living in the same probability space $(\widetilde{\Omega}, \widetilde{\mathcal{A}}, \widetilde{\mathbb{P}})$. In such a setting, $u^{(k)}$ and $J^{(k)}$ are realizations of the two random variables $\boldsymbol{U}^{(k)} = \varphi^{(k)}(\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(k)})$ and $\boldsymbol{J}^{(k)} = \psi^{(k)}(\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(k)})$. Using Theorem B.22, we deduce from (2.5) that the random variable $\boldsymbol{J}^{(k)}$ may be written as a conditional expectation:

$$\boldsymbol{J}^{(k)} = \mathbb{E}\Big(j\big(\varphi^{(k)}(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(k)}), \boldsymbol{W}\big) \;\Big|\; \boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(k)} \Big).$$

In this text, we simplify our notation and denote the space $(\widetilde{\Omega}, \widetilde{\mathcal{A}}, \widetilde{\mathbb{P}})$ by $(\Omega, \mathcal{A}, \mathbb{P})$. Remember that such a space has to be sufficiently big to contain an infinite-dimensional sample of $\boldsymbol{W}$.

In order to assess the quality of the approximated problem, we need to study the statistical properties of the estimators $\boldsymbol{U}^{(k)}$ and $\boldsymbol{J}^{(k)}$. For example, the bias of the approximated optimal cost is evaluated by computing $\mathbb{E}(\boldsymbol{J}^{(k)})$ and comparing it to the true optimal cost $J^\sharp$ of Problem (2.3). It is important to realize that the point we are interested in is the dependency of the solution w.r.t. the sampling. In this chapter, we mainly focus on the asymptotic properties of the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ (convergence and convergence rate).

2.3 Stochastic Gradient Method Overview

We now present the general method of the stochastic gradient algorithm, as well as convergence results related to the method.

2.3.1 Stochastic Gradient Algorithm

Algorithm

The *stochastic gradient algorithm* applies to Problem (2.3) and consists in devising a method where the optimization variable u evolves over the iterations using the gradient of j evaluated at successive realizations of the random variable $\boldsymbol{W}$, rather than using the gradient of J. Otherwise stated, one uses gradient iterations to perform the optimization task and, in the same process, to visit successive realizations of $\boldsymbol{W}$ with the purpose of evaluating the expectation as in a Monte Carlo technique.

Algorithm 2.1 (Stochastic Gradient Algorithm).

1. Pick up some $u^{(0)} \in U^{\mathrm{ad}}$ and choose a positive real sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$.
2. At iteration k, draw a realization $w^{(k+1)}$ of the random variable $\boldsymbol{W}$.
3. Compute the gradient of j w.r.t. u at point $(u^{(k)}, w^{(k+1)})$ and update $u^{(k+1)}$ by the formula: $u^{(k+1)} = \mathrm{proj}_{U^{\mathrm{ad}}}\Big(u^{(k)} - \epsilon^{(k)} \nabla_u j(u^{(k)}, w^{(k+1)})\Big)$.
4. Set $k = k + 1$ and go to step 2.

Algorithm 2.1 corresponds to the *numerical implementation* of the stochastic gradient method with a computer. The values $w^{(k)}$ involved in Algorithm 2.1 are drawn in such a way that the sequence $(w^{(1)}, \ldots, w^{(k)})$ is a realization of a k–sample of $\boldsymbol{W}$ (the reader is referred to Sect. B.7.4 for further details). This assumption is of paramount importance in order to ensure that Algorithm 2.1 converges towards the solution of

Problem (2.3). Note that we did not set a stopping test in the previous algorithm. This point is discussed in Sect. 2.6.

In order to study the convergence properties of such an algorithm, it is necessary to cast it in the adequate *probabilistic framework*. We thus consider a infinite-dimensional sample $\{\boldsymbol{W}^{(k)}\}_{k\in\mathbb{N}}$ of $\boldsymbol{W}$ (as defined in Sect. B.7.2). Step 3 of Algorithm 2.1 can be interpreted as an iterative relation involving random variables, namely

$$\boldsymbol{U}^{(k+1)} = \mathrm{proj}_{U^{\mathrm{ad}}}\left(\boldsymbol{U}^{(k)} - \epsilon^{(k)}\nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)})\right). \tag{2.6}$$

Each value $u^{(k)}$ computed by Algorithm 2.1 corresponds to a realization of the random variable $\boldsymbol{U}^{(k)}$. The projection in (2.6) is to be understood ω per ω.

Example: Estimation of an Expectation

Let us illustrate Algorithm 2.1 in the framework of statistical estimation, more precisely as an application of the Monte Carlo method. Let $\boldsymbol{W}$ be a real-valued integrable random variable defined on $(\Omega, \mathcal{A}, \mathbb{P})$, and suppose we want to compute an estimate of its expectation

$$\mathbb{E}(\boldsymbol{W}) = \int_\Omega \boldsymbol{W}(\omega)\,\mathrm{d}\mathbb{P}(\omega).$$

A way to do that is to draw a realization of a k-sample $(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(k)})$ of $\boldsymbol{W}$ and to compute the associated arithmetic mean. In terms of random variables, the estimator of the expectation associated with the k-sample is

$$\boldsymbol{U}^{(k)} = \frac{1}{k}\sum_{l=1}^{k} \boldsymbol{W}^{(l)}. \tag{2.7}$$

By the strong law of large numbers (Sect. B.7, Theorem B.27), the sequence of random variables $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ almost surely converges to $\mathbb{E}(\boldsymbol{W})$. From (2.7), we have that

$$\begin{aligned}
\boldsymbol{U}^{(k+1)} &= \frac{1}{k+1}\sum_{l=1}^{k} \boldsymbol{W}^{(l)} \;+\; \frac{\boldsymbol{W}^{(k+1)}}{k+1} \\
&= \frac{1}{k}\sum_{l=1}^{k} \boldsymbol{W}^{(l)} \;-\; \frac{1}{k+1}\left(\frac{1}{k}\sum_{l=1}^{k} \boldsymbol{W}^{(l)} - \boldsymbol{W}^{(k+1)}\right) \\
&= \boldsymbol{U}^{(k)} - \frac{1}{k+1}\left(\boldsymbol{U}^{(k)} - \boldsymbol{W}^{(k+1)}\right).
\end{aligned}$$

Using the notations $\epsilon^{(k)} = 1/(k+1)$ and $j(u, w) = \big(u - w\big)^2/2$, the last expression of $\boldsymbol{U}^{(k+1)}$ writes

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} - \epsilon^{(k)}\nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}). \tag{2.8}$$

Recalling that the expectation of $\boldsymbol{W}$ may be interpreted as the value which minimizes the dispersion of the random variable, namely

$$\mathbb{E}(\boldsymbol{W}) = \arg\min_{u\in\mathbb{R}} \frac{1}{2}\mathbb{E}\big((u-\boldsymbol{W})^2\big), \tag{2.9}$$

we conclude that the recursive form (2.8) of the Monte Carlo method exactly matches the stochastic gradient algorithm applied to the optimization problem (2.9). In the present case, U^{ad} is the whole space $\mathbb{R}$ so that $\mathrm{proj}_{U^{\mathrm{ad}}}(\cdot)$ is the identity function on $\mathbb{R}$.

This basic example makes it possible to enlighten some salient features of the stochastic gradient method.

- The step size $\epsilon^{(k)} = 1/(k+1)$ goes to zero as k goes to infinity, whereas the step size may be constant for deterministic optimization algorithms. Note however that $\epsilon^{(k)}$ goes to zero "not too fast", that is,

$$\sum_{k\in\mathbb{N}} \epsilon^{(k)} = +\infty.$$

  Of c ourse, it would be awkward for the series $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$ to be convergent, because it should be clear that the algorithm would converge to a limit which depends on the initial point $u^{(0)}$ and on the sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$ itself. For example, consider the case $U^{\mathrm{ad}} = \mathbb{R}$ and $j(u,w) = |u|$ (hence $\nabla_u j(u,w) = -1$ for $u < 0$). Starting from $u^{(0)} < -1$ with step sizes $\epsilon^{(k)} = 1/2^{k+1}$, Algorithm 2.1 leads to

$$u^{(k+1)} = u^{(0)} + \sum_{l=1}^{k+1} \frac{1}{2^l}, \quad \text{so that} \quad \lim_{k\to+\infty} u^{(k)} = u^{(0)} + 1 < 0,$$

  whereas the solution of the optimization problem $\min_{u\in\mathbb{R}} |u|$ is $u^\sharp = 0$.
- The underlying convergence notion in this example is the one of the strong law of large numbers, that is, almost sure convergence. It is thus reasonable to expect such a convergence for the stochastic gradient algorithm (rather than a weaker notion as convergence in distribution or convergence in probability).
- As the central limit theorem applies to this example (Theorem B.28), we can expect a similar result for the rate of convergence of the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ generated by the stochastic gradient algorithm.

Probabilistic Considerations

Iteration k of the stochastic gradient method (2.6) can be represented by the general relation

$$\boldsymbol{U}^{(k+1)} = \mathcal{R}^{(k)}\big(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}\big). \tag{2.10}$$

We assume that the random variable $\boldsymbol{U}^{(0)}$ is constant, equal to $u^{(0)} \in U^{\mathrm{ad}}$, and that the mappings $\mathcal{R}^{(k)}$ are measurable.

- Let $\mathcal{F}^{(k)}$ be the subfield generated by the k-sample $(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(k)})$:

$$\mathcal{F}^{(0)} = \{\emptyset, \Omega\}\,, \quad \mathcal{F}^{(k)} = \sigma\big(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(k)}\big).$$

  The sequence $\{\mathcal{F}^{(k)}\}_{k\in\mathbb{N}}$ is a filtration, that is, $\mathcal{F}^{(k)} \subset \mathcal{F}^{(k+1)}$.
- By induction on (2.10), $\boldsymbol{U}^{(k)}$ is driven by $(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(k)})$. The random variable $\boldsymbol{U}^{(k)}$ is thus $\mathcal{F}^{(k)}$-measurable for all k.
- Defining the function $\varphi^{(k)}$ as

$$\varphi^{(k)}(u) = \mathbb{E}\big(\mathcal{R}^{(k)}(u, \boldsymbol{W})\big),$$

  using the fact that the random variables $\boldsymbol{W}^{(k)}$ are independent and that $\boldsymbol{U}^{(k)}$ is $\mathcal{F}^{(k)}$-measurable, one obtains from Theorem B.22 that

$$\begin{aligned}\mathbb{E}\big(\boldsymbol{U}^{(k+1)} \,\big|\, \mathcal{F}^{(k)}\big) &= \mathbb{E}\big(\mathcal{R}^{(k)}(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}) \,\big|\, \mathcal{F}^{(k)}\big)\\ &= \varphi^{(k)}\big(\boldsymbol{U}^{(k)}\big),\end{aligned}$$

  that is, for almost every $\omega \in \Omega$,

$$\mathbb{E}\big(\boldsymbol{U}^{(k+1)} \,\big|\, \mathcal{F}^{(k)}\big)(\omega) = \int_{\Omega} \mathcal{R}^{(k)}\big(\boldsymbol{U}^{(k)}(\omega), \boldsymbol{W}(\omega')\big)\, \mathrm{d}\mathbb{P}(\omega').$$

  The conditional expectation of $\boldsymbol{U}^{(k+1)}$ given $\mathcal{F}^{(k)}$ thus consists merely of a standard expectation.
- As observed in the previous example, the candidate convergence notion for studying (2.10) is the almost sure convergence. Note that the almost sure convergence of the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ towards a constant $u^\sharp$ has the following intuitive meaning: almost every run of Algorithm 2.1 produces a sequence $\{u^{(k)}\}_{k\in\mathbb{N}}$ converging to $u^\sharp$.

2.3.2 Connection with Stochastic Approximation

A classical problem considered in the Stochastic Approximation (SA) framework is to determine the zero of a function h using noisy evaluations of this function. Let $\mathbb{U}$ be the finite-dimensional Hilbert space $\mathbb{R}^n$. We consider a mapping $h : \mathbb{U} \to \mathbb{U}$, and we assume that the observation of $h(u)$ is perturbed by an additive random variable $\boldsymbol{\xi}$. The standard Stochastic Approximation algorithm consists in determining the zero of h by the following recursive formula:[2]

[2]The positive sign in front of $\epsilon^{(k)}$ in the update formula (2.11) is explained later on.

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} + \epsilon^{(k)}\Big(h(\boldsymbol{U}^{(k)}) + \boldsymbol{\xi}^{(k+1)}\Big). \tag{2.11}$$

This algorithm is strongly related to the stochastic gradient algorithm. Indeed, consider the minimization problem (2.3) and assume that the admissible set U^{ad} is equal to $\mathbb{U}$. The projection onto U^{ad} is, accordingly, the identity operator, and the k-th iteration of the stochastic gradient algorithm writes

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} - \epsilon^{(k)}\nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}). \tag{2.12}$$

Defining the mapping h and the random variables $\boldsymbol{\xi}^{(k+1)}$ as

$$h(u) = -\nabla J(u)\ , \tag{2.13a}$$

$$\boldsymbol{\xi}^{(k+1)} = \nabla J(\boldsymbol{U}^{(k)}) - \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}), \tag{2.13b}$$

the stochastic gradient recursion (2.12) is identical to (2.11). Note that the problem of finding a point $u^\sharp \in \mathbb{U}$ such that $h(u^\sharp) = 0$ is equivalent to solving $\nabla J(u^\sharp) = 0$, a necessary condition for $u^\sharp$ to be a solution of Problem (2.2).

In the next two paragraphs, we deal with the Stochastic Approximation formulation and we present two important results about the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ generated by (2.11). In such a setting, a filtration $\{\mathcal{F}^{(k)}\}_{k\in\mathbb{N}}$ is given, and $\{\boldsymbol{\xi}^{(k)}\}_{k\in\mathbb{N}}$ is a sequence of $\mathbb{U}$-valued random variables. The random variable $\boldsymbol{U}^{(0)}$ is used to initiate the recursion (2.11).

Robbins-Monro Theorem

Here we focus on the convergence of the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ of random variables generated by (2.11). According to the observations made about the example considered in Sect. 2.3.1, the step sizes $\epsilon^{(k)}$ should be positive and should go to zero "not too fast". We first specify such a behavior.

Definition 2.2 A positive real sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$ is a *σ-sequence* if it satisfies the two properties

$$\sum_{k\in\mathbb{N}} \epsilon^{(k)} = +\infty, \quad \sum_{k\in\mathbb{N}} \left(\epsilon^{(k)}\right)^2 < +\infty.$$

We make the following assumptions on the different components involved in (2.11).

Assumptions 2.3

1. The random variable $\boldsymbol{U}^{(0)}$ is $\mathcal{F}^{(0)}$-measurable.
2. The mapping $h\ :\ \mathbb{U} \longrightarrow \mathbb{U}$ is continuous, such that
   - $\exists\ u^\sharp \in \mathbb{R}^n,\ h(u^\sharp) = 0$ and $\left\langle h(u)\ , u - u^\sharp\right\rangle < 0, \forall u \neq u^\sharp$;
   - $\exists\ a > 0,\ \forall u \in \mathbb{R}^n,\ \|h(u)\|^2 \leq a\big(1 + \|u\|^2\big)$.
3. The random variable $\boldsymbol{\xi}^{(k)}$ is $\mathcal{F}^{(k)}$-measurable for all k, and

- $\mathbb{E}\big(\boldsymbol{\xi}^{(k+1)} \mid \mathcal{F}^{(k)}\big) = 0$,
- $\exists\, d > 0,\ \mathbb{E}\big(\|\boldsymbol{\xi}^{(k+1)}\|^2 \mid \mathcal{F}^{(k)}\big) \leq d\big(1 + \|\boldsymbol{U}^{(k)}\|^2\big)$.

4. The sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$ is a *σ-sequence*.

Remark 2.4 Assumption 2.3-2 implies that $u^\sharp$ is the unique zero of h. ◊

Remark 2.5 The stepsize $\epsilon^{(k)}$ could be considered as the realization of a random variable $\boldsymbol{\epsilon}^{(k)}$ satisfying Definition 2.2 $\mathbb{P}$-a.s.. It would then be necessary to add the assumption that $\boldsymbol{\epsilon}^{(k)}$ is measurable with respect to $\mathcal{F}^{(k)}$. ◊

Theorem 2.6 below is a particular case of the standard Robbins-Monro theorem presented in [129] or in [60].

Theorem 2.6 *Under Assumptions 2.3, the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ of random variables generated by (2.11) almost surely converges to $u^\sharp$.*

For a proof, see [60, Sect. 1.4].

Let us detail the connection between the assumptions we may formulate about the initial problem (2.3) and the assumptions of Theorem 2.6. We assume that the σ-field $\mathcal{F}^{(k)}$ is generated by $\big(\boldsymbol{W}^{(0)}, \ldots, \boldsymbol{W}^{(k)}\big)$, so that we deduce from (2.13) that $\boldsymbol{\xi}^{(k)}$ is $\mathcal{F}^{(k)}$-measurable. We assume that the function j is strictly convex, coercive, continuously differentiable w.r.t. u and measurable w.r.t. w. Then J is strictly convex, coercive and continuously differentiable. The first part of Assumption 2.3-2 is related to these assumptions which ensure the existence and uniqueness of the solution of Problem (2.3), whereas the first part of Assumption 2.3-3 is an immediate consequence of (2.13). As for the second parts of Assumptions 2.3-2 and 2.3-3, they may be connected with a *linearly bounded gradient* (LBG) assumption on j, that is,

$$\exists c_1 > 0,\ \ c_2 > 0,\ \ \forall u \in \mathbb{R}^n,\ \ \forall w \in \mathbb{W},\ \ \|\nabla_u j(u, w)\| \leq c_1 \|u\| + c_2,$$

which implies that (hint: use $(a+b)^2 \leq 2(a^2+b^2)$)

$$\exists\, c_3 > 0,\ \ c_4 > 0,\ \ \forall u \in \mathbb{R}^n,\ \ \forall w \in \mathbb{W},\ \ \|\nabla_u j(u, w)\|^2 \leq c_3 \|u\|^2 + c_4,$$
$$\|\nabla J(u)\|^2 \leq c_3 \|u\|^2 + c_4.$$

These assumptions about the cost function j are natural in the convex optimization context. In Sect. 2.4, we give a more general convergence result concerning the stochastic gradient algorithm.

Remark 2.7 Theorem 2.6 can be extended to more general situations.

- As in Algorithm 2.1, a projection operator can be added to (2.11):

$$\boldsymbol{U}^{(k+1)} = \mathrm{proj}_{U^{\mathrm{ad}}}\Big(\boldsymbol{U}^{(k)} + \epsilon^{(k)}\big(h(\boldsymbol{U}^{(k)}) + \boldsymbol{\xi}^{(k+1)}\big)\Big).$$

Here U^{ad} is a non empty closed convex subset of $\mathbb{U}$.

- A "small" additional term $\boldsymbol{R}^{(k+1)}$ can be added to (2.11):

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} + \epsilon^{(k)}\big(h(\boldsymbol{U}^{(k)}) + \boldsymbol{\xi}^{(k+1)} + \boldsymbol{R}^{(k+1)}\big).$$

  Such a term may be interpreted as a bias on $h(u)$ which vanishes asymptotically, as considered in the Kiefer-Wolfowitz algorithm [93].

The reader is referred to [54, 59] for further details. ◊

Rate of Convergence

We now recall a central limit type theorem for the stochastic approximation method, that is, a result about the asymptotic normality of the random variables $\boldsymbol{U}^{(k)}$ generated by (2.11), together with an estimation of the rate of convergence of such an algorithm. Here we need to be more specific about the notion of σ-sequence and we give the following definition.

Definition 2.8 A positive real sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$ is a $\sigma(\alpha, \beta, \gamma)$-*sequence* if it is such that

$$\epsilon^{(k)} = \frac{\alpha}{k^\gamma + \beta},$$

with $\alpha > 0$, $\beta \geq 0$ and $1/2 < \gamma \leq 1$.

An immediate consequence of this definition is that a $\sigma(\alpha, \beta, \gamma)$-sequence is also a σ-sequence.

We retain Assumptions 2.3 to ensure that the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ almost surely converges to $u^\sharp$, and we make the following additional assumptions.

Assumptions 2.9

1. The mapping h is continuously differentiable and has the following expression in a neighborhood of $u^\sharp$

$$h(u) = -H(u - u^\sharp) + \mathrm{O}(\|u - u^\sharp\|^2),$$

   where H is a symmetric positive-definite matrix.[3]
2. The sequence $\big\{\mathbb{E}\big(\boldsymbol{\xi}^{(k+1)}(\boldsymbol{\xi}^{(k+1)})^\top \,\big|\, \mathcal{F}^{(k)}\big)\big\}_{k\in\mathbb{N}}$ of conditional covariance matrices almost surely converges to a symmetric positive-definite matrix Γ.
3. There exists $\delta > 0$ such that $\sup_{k\in\mathbb{N}} \mathbb{E}\big(\|\boldsymbol{\xi}^{(k+1)}\|^{2+\delta} \,\big|\, \mathcal{F}^{(k)}\big) < +\infty$.
4. The sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$ is a $\sigma(\alpha, \beta, \gamma)$-sequence.
5. The square matrix $(H - \lambda I)$ is positive-definite, λ being defined as

$$\lambda = \begin{cases} 0 & \text{if } \gamma < 1, \\ \dfrac{1}{2\alpha} & \text{if } \gamma = 1. \end{cases}$$

[3] The symbol O corresponds to the "Big-O" notation: $f(x) = \mathrm{O}\big(g(x)\big)$ as $x \to x_0$ if and only if there exist a positive constant α and a neighborhood V of x_0 such that $|f(x)| \leq \alpha\,|g(x)|$, $\forall x \in V$.

Remark 2.10 If we refer back to the initial problem (2.3) where $h = -\nabla J$, we notice that H is the Hessian matrix of J at $u^\sharp$

$$H = \nabla^2 J(u^\sharp).$$

Moreover, since $\mathbb{E}\big(\nabla_u j(u^\sharp, \boldsymbol{W})\big) = 0$, the matrix Γ introduced in Assumption 2.9-2 is equal to the covariance matrix of $\nabla_u j$ evaluated at $u^\sharp$

$$\Gamma = \mathbb{E}\Big(\nabla_u j(u^\sharp, \boldsymbol{W})\big(\nabla_u j(u^\sharp, \boldsymbol{W})\big)^\top\Big). \qquad \Diamond$$

The rate of convergence of the random variables $\boldsymbol{U}^{(k)}$ generated by (2.11) is given by Theorem 2.11. This theorem is a particular case of the one presented in [59].

Theorem 2.11 *Under Assumptions 2.3 and 2.9, the sequence of random variables* $\big\{(1/\sqrt{\epsilon^{(k)}})(\boldsymbol{U}^{(k)} - u^\sharp)\big\}_{k\in\mathbb{N}}$ *converges in law*[4] *to a centered gaussian distribution with covariance matrix* Σ*, that is,*

$$\frac{1}{\sqrt{\epsilon^{(k)}}}\Big(\boldsymbol{U}^{(k)} - u^\sharp\Big) \xrightarrow{\mathcal{D}} \mathcal{N}\big(0, \Sigma\big), \tag{2.14}$$

in which Σ *is the solution of the so-called Lyapunov equation*

$$\big(H - \lambda I\big)\Sigma + \Sigma\big(H - \lambda I\big) = \Gamma. \tag{2.15}$$

For a proof, see [59, Chap. 4]; see also [54] for a detailed step-by-step proof.

Remark 2.12 As already mentioned in Remark 2.7, the Robbins-Monro Theorem 2.6 remains valid when one adds a projection operator to (2.11). This is not true for Theorem 2.11 which only deals with unconstrained problems ($U^{\mathrm{ad}} = \mathbb{U}$), or at least with problems such that $u^\sharp$ belongs to the interior of the set U^{ad}. $\Diamond$

For the sake of completeness, we recall the characterization of solutions of Lyapunov equations. The following theorem can be found in [92, Theorem 4.6].

Proposition 2.13 *Let* H *be a positive-definite matrix and* Γ *be a symmetric positive-definite matrix. Then, the Lyapunov equation*

$$H\Sigma + \Sigma H^\top = \Gamma \tag{2.16}$$

admits a unique symmetric positive-definite solution Σ *given by:*

$$\Sigma = \int_0^{+\infty} e^{-tH}\Gamma e^{-tH^\top}\,\mathrm{d}t. \tag{2.17}$$

[4] See Sect. B.3.4 for this convergence notion and for the associated notation $\xrightarrow{\mathcal{D}}$.

Remark 2.14 This result remains true if Γ is a nonnegative-definite matrix: then, the matrix Σ given by (2.17) is a nonnegative-definite matrix, and is the solution of Eq. (2.16). ◊

In order to be more accurate about the convergence rate given by Theorem 2.11, let us examine the respective influence of the coefficients α, β and γ entering the expression of step sizes $\epsilon^{(k)}$ defined in Assumption 2.9-4.

- The convergence result of Theorem 2.11 can be rephrased as

$$k^{\frac{\gamma}{2}}\left(\boldsymbol{U}^{(k)} - u^{\sharp}\right) \xrightarrow{\mathcal{D}} \mathcal{N}\big(0, \alpha\Sigma\big), \tag{2.18}$$

  so that the coefficient β has in fact no influence on the convergence rate. The way in which β alters the transient behavior of the algorithm is explained in Sect. 2.6.2.
- It follows from (2.18) that the optimal choice for γ, that is, the value achieving the greatest convergence rate in (2.14), is $\gamma = 1$. We recover the "classical" rate $1/\sqrt{k}$ provided by a Monte Carlo estimator.

The next question is: which choice of α induces a covariance matrix $\alpha\Sigma$ in (2.18) as small as possible (in the cone of positive-definite matrices)? This problem is addressed in Sect. 2.5. Observe, for the time being, that the simplistic reasoning which consists in taking α as small as possible in order to minimize the covariance in (2.18) does not hold. Indeed, using the optimal value $\gamma = 1$, the solution Σ of the Lyapunov equation (2.15) depends on λ and hence on α, so that the covariance matrix $\alpha\Sigma$ is not a *linear* nor a *monotonic* function of α. For example, in the scalar case ($n = 1$), H and Γ are real numbers and the solution of (2.15) is

$$\Sigma = \frac{\alpha\Gamma}{2\alpha H - 1}.$$

Minimizing $\alpha\Sigma$ w.r.t. α leads to the optimal value $\alpha^{\sharp} = 1/H$, which is compatible with the condition $\alpha > 1/2H$ imposed by Assumption 2.9-5.

2.4 Convergence Analysis

We now consider a generalization of the stochastic gradient Algorithm 2.1 derived from the so-called Auxiliary Problem Principle, and we give a convergence result for this generalized algorithm.

2.4.1 Auxiliary Problem Principle

Consider the following optimization problem

$$\min_{u \in U^{\mathrm{ad}}} J(u). \tag{2.19}$$

Let $u^\sharp \in U^{\mathrm{ad}}$ be a solution of this problem. We recall (see Theorem A.10) that the associated optimality condition writes

$$\left\langle \nabla J(u^\sharp)\,,\, u - u^\sharp \right\rangle \geq 0, \quad \forall u \in U^{\mathrm{ad}}. \tag{2.20}$$

In the deterministic framework, the *Auxiliary Problem Principle*[5] (**APP**) consists in replacing Problem (2.19) by a sequence of auxiliary problems indexed by $k \in \mathbb{N}$. Let K be a real-valued differentiable function defined on $\mathbb{U}$ and let ϵ be a positive constant. At iteration k, knowing $u^{(k)} \in U^{\mathrm{ad}}$, consider the auxiliary problem

$$\min_{u \in U^{\mathrm{ad}}} K(u) + \left\langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)})\,,\, u \right\rangle. \tag{2.21}$$

Its solution $u^{(k+1)}$ is used to formulate the $(k+1)$-th auxiliary problem.

The interest of such a principle lies in the fact that the resolution of the auxiliary problem (2.21) may be much easier to obtain than the solution of the initial problem (2.19). Namely, the function K appearing in (2.21) is part of the algorithm design (K is called a *core*). The choice of K being subject to rather mild conditions, one can take advantage of a proper choice in order to obtain many special features for Problem (2.21). The main properties of the Auxiliary Principle Problem are examined hereafter.

- **APP** is consistent. Assuming that the sequence of solutions $\{u^{(k)}\}_{k \in \mathbb{N}}$ converges to some $u^\sharp$ and taking the limit in the optimality condition of Problem (2.21)

$$\left\langle \nabla K(u^{(k+1)}) + \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)})\,,\, u - u^{(k+1)} \right\rangle \geq 0, \quad \forall u \in U^{\mathrm{ad}},$$

  we obtain the optimality conditions (2.20), up to a factor ϵ, by cancellation of the gradients of K (we assume that ∇K is continuous at $u^\sharp$). This shows that $u^\sharp$ is a solution of Problem (2.19) at least in the convex case.
- **APP** encompasses numerous classical optimization algorithms. For example, using a quadratic core $K(u) = (1/2)\,\|u\|^2$, Problem (2.21) writes

$$\min_{u \in U^{\mathrm{ad}}} \frac{1}{2}\,\|u\|^2 + \left\langle \epsilon \nabla J(u^{(k)}) - u^{(k)}\,,\, u \right\rangle,$$

[5] See [39] for a reference about the Auxiliary Problem Principle.

and its solution has the following closed-form expression

$$u^{(k+1)} = \mathrm{proj}_{U^{\mathrm{ad}}}\left(u^{(k)} - \epsilon \nabla J(u^{(k)})\right).$$

We thus obtain the well-known projected gradient algorithm.

- **APP** allows for decomposition. Assume that the space $\mathbb{U}$ is the Cartesian product of N spaces:

$$\mathbb{U} = \prod_{i=1}^{N} \mathbb{U}_i.$$

Assume, moreover, that the admissible set U^{ad} is the Cartesian product of N sets $(U_1^{\mathrm{ad}}, \ldots, U_N^{\mathrm{ad}})$, with $U_i^{\mathrm{ad}} \subset \mathbb{U}_i$. That is, the constraint $u \in U^{\mathrm{ad}}$ is equivalent to the set of N independent constraints $u_i \in U_i^{\mathrm{ad}}$ for the components u_i of u. If we choose a core function K additive according to that decomposition of u, namely $K(u) = \sum_{i=1}^{N} K_i(u_i)$, Problem (2.21) becomes

$$\min_{(u_1,\ldots,u_N)\in U_1^{\mathrm{ad}}\times\cdots\times U_N^{\mathrm{ad}}} \sum_{i=1}^{N} \left(K_i(u_i) + \left\langle \epsilon \nabla_{u_i} J(u^{(k)}) - \nabla K_i(u_i^{(k)}) \, , u_i \right\rangle\right).$$

This problem splits up into N independent optimization subproblems, the ith subproblem being

$$\min_{u_i \in U_i^{\mathrm{ad}}} K_i(u_i) + \left\langle \epsilon \nabla_{u_i} J(u^{(k)}) - \nabla K_i(u_i^{(k)}) \, , u_i \right\rangle.$$

The reader is referred to [39–41] for a detailed description of the **APP** (see also the more recent lecture notes [38]).

2.4.2 Stochastic Auxiliary Problem Principle Algorithm

Let us consider the optimization problem (2.3), that we repeat here for convenience

$$\min_{u \in U^{\mathrm{ad}}} J(u), \tag{2.22}$$

with $J(u) = \mathbb{E}\big(j(u, \boldsymbol{W})\big)$. In order to mix the ideas of the Auxiliary Problem Principle and of the Stochastic Gradient Method, we first replace Problem (2.22) by the associated sequence of auxiliary problems, namely

$$\min_{u \in U^{\mathrm{ad}}} K(u) + \left\langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}) \, , u \right\rangle.$$

Then, in each auxiliary problem, we replace the gradient of J by the gradient of j evaluated at sampled realizations of $\boldsymbol{W}$; moreover, the "large" (constant) step size ϵ must be replaced by "small" (going to zero as index k goes to infinity) step sizes $\epsilon^{(k)}$. The k-th instance of the stochastic auxiliary problem is thus

$$\min_{u \in U^{\mathrm{ad}}} K(u) + \left\langle \epsilon^{(k)} \nabla_u j(u^{(k)}, w^{(k+1)}) - \nabla K(u^{(k)}) \, , u \right\rangle, \tag{2.23}$$

$w^{(k+1)}$ being a realization of the random variable $\boldsymbol{W}$. This results in the following algorithm.

Algorithm 2.15 (Stochastic APP Algorithm).

1. Pick up some $u^{(0)} \in U^{\mathrm{ad}}$ and choose a positive real sequence $\{\epsilon^{(k)}\}_{k \in \mathbb{N}}$.
2. At iteration k, draw a realization $w^{(k+1)}$ of the random variable $\boldsymbol{W}$.
3. Update $u^{(k+1)}$ by solving the auxiliary problem (2.23):
   $u^{(k+1)} \in \arg\min_{u \in U^{\mathrm{ad}}} K(u) + \left\langle \epsilon^{(k)} \nabla_u j(u^{(k)}, w^{(k+1)}) - \nabla K(u^{(k)}) \, , u \right\rangle.$
4. Set $k = k + 1$ and go to step 2.

As already pointed out when devising Algorithm 2.1, the values $w^{(k)}$ involved in Algorithm 2.15 are drawn in such a way that the sequence $(w^{(1)}, \ldots, w^{(k)})$ is a realization of a k–sample of $\boldsymbol{W}$.

Remark 2.16 With the choice $K(u) = \|u\|^2/2$, the auxiliary problem (2.23) becomes

$$\min_{u \in U^{\mathrm{ad}}} \frac{1}{2} \|u\|^2 + \left\langle \epsilon^{(k)} \nabla_u j(u^{(k)}, w^{(k+1)}) - u^{(k)} \, , u \right\rangle.$$

Its unique solution $u^{(k+1)}$ is given by

$$u^{(k+1)} = \mathrm{proj}_{U^{\mathrm{ad}}} \left(u^{(k)} - \epsilon^{(k)} \nabla_u j(u^{(k)}, w^{(k+1)}) \right).$$

This relation precisely corresponds to the stochastic gradient iteration of Algorithm 2.1. ◊

We now focus on the convergence analysis of the stochastic APP Algorithm 2.15. We restrict ourselves to the differentiable case, but everything remains valid for subdifferentiable functions (see [45, 47] for further details).

2.4.3 Convergence Theorem

As in Sect. 2.3, we consider the stochastic APP Algorithm 2.15 in terms of random variables. Let $\{\boldsymbol{W}^{(k)}\}_{k \in \mathbb{N}}$ be an infinite dimensional sample of $\boldsymbol{W}$. The auxiliary problem at iteration k is

$$\min_{u \in U^{\mathrm{ad}}} K(u) + \left\langle \epsilon^{(k)} \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}) - \nabla K(\boldsymbol{U}^{(k)}) \, , u \right\rangle, \tag{2.24}$$

and the minimization in (2.24) is to be understood ω per ω. Assume that the set-valued random mapping corresponding to the arg min of Problem (2.24) admits a measurable selection $\boldsymbol{U}^{(k+1)}$ (this is justified in the proof of the following theorem). The convergence properties of the sequence of random variables $\{\boldsymbol{U}^{(k)}\}_{k \in \mathbb{N}}$ generated by (2.24) and the connection with the initial problem (2.22) are stated in the following theorem.

Theorem 2.17 *We make the following assumptions.*

1. U^{ad} *is a non empty closed convex subset of a Hilbert space* $\mathbb{U}$.
2. *The function* $j : \mathbb{U} \times \mathbb{W} \to \mathbb{R}$ *is a normal integrand,*[6] *and* $\mathbb{E}\big(j(u, \boldsymbol{W})\big)$ *exists for all* $u \in U^{\mathrm{ad}}$.
3. *The function* $j(\cdot, w) : \mathbb{U} \to \mathbb{R}$ *is proper, convex, lower semi-continuous and differentiable on an open subset containing* U^{ad}, *for all* $w \in \mathbb{W}$.[7]
4. *The function* $j(\cdot, w)$ *has linearly bounded gradients (LBG), uniformly in* w*:*

$$\exists c_1 > 0, \ \exists c_2 > 0, \ \forall w \in \mathbb{W}, \ \forall u \in U^{\mathrm{ad}}, \ \|\nabla_u j(u, w)\| \leq c_1 \|u\| + c_2.$$

5. *The function* J *is coercive on* U^{ad}.[8]
6. *The core function* K *is proper, strongly convex with modulus* b*, lower semi-continuous and differentiable on an open subset containing* U^{ad}.
7. *The sequence* $\{\epsilon^{(k)}\}_{k \in \mathbb{N}}$ *is a* σ*-sequence.*

Then the following conclusions hold true.

1. *Problem (2.22) has a non empty set of solutions* $U^\sharp$.
2. *Problem (2.24) has a unique solution* $\boldsymbol{U}^{(k+1)}$.
3. *The sequence of random variables* $\{J(\boldsymbol{U}^{(k)})\}_{k \in \mathbb{N}}$ *almost surely converges to* $\min_{u \in U^{\mathrm{ad}}} J(u)$.
4. *The sequence of random variables* $\{\boldsymbol{U}^{(k)}\}_{k \in \mathbb{N}}$ *is almost surely bounded, and every cluster point of a realization of this sequence belongs to the optimal set* $U^\sharp$.

At last, if J *is strongly convex,* $U^\sharp$ *is a singleton* $\{u^\sharp\}$ *and the sequence* $\{\boldsymbol{U}^{(k)}\}_{k \in \mathbb{N}}$ *almost surely converges to the unique solution* $u^\sharp$ *of Problem (2.22).*

Proof The proof of Theorem 2.4.3 is rather long and technical. This is the reason why it has been postponed to the end of the present chapter, and we just give here a sketch of the proof. The proof of the first two statements is based on classical theorems in the field of convex optimization. The property that the solution $\boldsymbol{U}^{(k+1)}$ of Problem (2.24) is a random variable (hence, measurable) is a consequence of the

[6] See Definition 8.22. This implies that $j(u, \boldsymbol{W}) : \Omega \to \mathbb{R}$ is measurable $\forall u \in U^{\mathrm{ad}}$.

[7] Note that the semi-continuity of $j(\cdot, w)$ stems from the fact that j is a normal integrand.

[8] See (A.5) for the meaning of this term.

fact that the criterion j is a normal integrand. The proof of the last two statements involves four steps.

1. **Select a Lyapunov function** Λ. Let $u^\sharp \in U^\sharp$ be a solution of (2.22) and consider the function
$$\Lambda(u) = K(u^\sharp) - K(u) - \left\langle \nabla K(u)\,, u^\sharp - u \right\rangle.$$
From the strong convexity of K, we have that
$$\left\| \boldsymbol{U} - u^\sharp \right\|^2 \leq \frac{2}{b} \Lambda(\boldsymbol{U}), \ \ \mathbb{P}\text{-a.s..} \tag{2.25}$$

2. **Bound from above the variation of** Λ. The optimality conditions for the auxiliary problem (2.24) evaluated at $\boldsymbol{U} = \boldsymbol{U}^{(k)}$ together with the strong convexity of K imply that
$$\left\| \boldsymbol{U}^{(k+1)} - \boldsymbol{U}^{(k)} \right\| \leq \frac{\epsilon^{(k)}}{b} \left\| \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}) \right\|, \ \ \mathbb{P}\text{-a.s..} \tag{2.26}$$
From the LBG assumption and using (2.25), we obtain that there exist positive constants α and β such that
$$\left\| \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}) \right\|^2 \leq \alpha \Lambda(\boldsymbol{U}^{(k)}) + \beta, \ \ \mathbb{P}\text{-a.s..} \tag{2.27}$$
All these inequalities are combined to obtain the following inequality:
$$\mathbb{E}\left(\Lambda(\boldsymbol{U}^{(k+1)}) \,\middle|\, \mathcal{F}^{(k)} \right) \leq (1+\alpha^{(k)}) \Lambda(\boldsymbol{U}^{(k)}) + \beta^{(k)} - \epsilon^{(k)} \left(J(\boldsymbol{U}^{(k)}) - J(u^\sharp) \right), \ \ \mathbb{P}\text{-a.s..} \tag{2.28}$$
with $\alpha^{(k)} = (\alpha/b)(\epsilon^{(k)})^2$ and $\beta^{(k)} = (\beta/b)(\epsilon^{(k)})^2$.

3. **Prove the convergence**. A straightforward application of the Robbins-Siegmund Theorem 2.27 shows that the sequence $\left\{ \Lambda(\boldsymbol{U}^{(k)}) \right\}_{k\in\mathbb{N}}$ almost surely converges to a finite random variable $\boldsymbol{\Lambda}^\infty$, and that the series $\sum \epsilon^{(k)} \left(J(\boldsymbol{U}^{(k)}) - J(u^\sharp) \right)$ almost surely converges.

4. **Characterize the sequence limits**. The convergence of $\left\{ \Lambda(\boldsymbol{U}^{(k)}) \right\}_{k\in\mathbb{N}}$ together with (2.27) imply that the sequence $\left\{ \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}) \right\}_{k\in\mathbb{N}}$ is almost surely finite. Thank to (2.26), Lemma 2.28 applies, so that the sequence $\{J(\boldsymbol{U}^{(k)})\}_{k\in\mathbb{N}}$ almost surely converges to $J(u^\sharp)$. From (2.25), we obtain that the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ is also almost surely finite: by a compactness argument, there exist subsequences converging to elements belonging to the set $U^\sharp$. □

2.4.4 Conclusions

We have given a general convergence theorem for the stochastic Auxiliary Problem Principle method. This theorem encompasses the standard stochastic gradient algorithm (obtained using the core function $K(u) = \|u\|^2/2$), as well as the so-called matrix-gain algorithm (the core function being in this case $K(u) = \langle u\,,\, Au\rangle/2$, A being a positive definite matrix).

From a theoretical point of view, Theorem 2.17 has been proved under natural assumptions. As a matter of fact, the convexity and differentiability assumptions are standard in the framework of convex optimization. Note moreover that, even if an explicit convexity property is not required in the Robbins-Monro Theorem 2.6, Assumption 2.3-2 plays in fact a very similar role.

As far as decomposition is concerned, the Auxiliary Problem Principle opens this possibility as a way to solve large stochastic optimization problems of the type (2.3).

2.5 Efficiency and Averaging

In this section we focus on the convergence rate of the stochastic gradient method. We use the setting considered in Sect. 2.3.2 for a *non constrained* stochastic optimization problem, that is,

$$\min_{u\in\mathbb{R}^n} J(u), \tag{2.29}$$

with $J(u) = \mathbb{E}\big(j(u, \boldsymbol{W})\big)$. Using a $\sigma(\alpha, \beta, \gamma)$-sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$, that is, step sizes $\epsilon^{(k)}$ of the form $\alpha/(k^\gamma + \beta)$, we know from Theorem 2.11 that

$$k^{\frac{\gamma}{2}}\Big(\boldsymbol{U}^{(k)} - u^\sharp\Big) \xrightarrow{\mathcal{D}} \mathcal{N}\big(0, \alpha\Sigma\big).$$

It has already been noted that the choice $\gamma = 1$ leads to the largest convergence rate. We want now to improve the convergence speed by minimizing the covariance matrix $\alpha\Sigma$ w.r.t. the symmetric positive-definite matrix cone.

2.5.1 Stochastic Newton Algorithm

In deterministic optimization, it is well-known that pre-multiplying the gradient of the function to be optimized by a (cleverly chosen) matrix can significantly improve the algorithm behavior. For example, using the inverse of the Hessian matrix leads to the Newton algorithm, which yields a (local) quadratic convergence rate whereas the convergence rate of the gradient method is only linear. It is of course unrealistic to expect such a nice result in the field of stochastic approximation because the step

size $\epsilon^{(k)}$ goes to zero as k goes to infinity, but we can expect some improvement of the method by a proper preconditioning of the gradient.

In order to apply this idea to the stochastic gradient method, we choose a symmetric positive-definite matrix A of dimension n. The step sizes $\epsilon^{(k)}$ are then built using the optimal choice $\gamma = 1$ and replacing the *scalar* gain α by the *matrix* gain A. Using these choices, the stochastic gradient iteration (2.12) becomes

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} - \frac{1}{k+\beta} A \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}),$$

which in the Stochastic Approximation setting (2.11)—(2.13) writes

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} + \frac{1}{k+\beta}\Big(Ah(\boldsymbol{U}^{(k)}) + A\boldsymbol{\xi}^{(k+1)}\Big). \tag{2.30}$$

The results stated in Sect. 2.3.2 are thus available, provided that we make use of modified data, namely a mapping Ah, noises $A\boldsymbol{\xi}^{(k)}$ and step sizes $1/(k+\beta)$. In the context of (2.30), Assumption 2.9-5 reads: $AH - I/2$ is a positive-definite matrix. Theorem 2.11 applies, so that the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ generated by (2.30) is such that

$$\sqrt{k}\Big(\boldsymbol{U}^{(k)} - u^\sharp\Big) \xrightarrow{\mathcal{D}} \mathcal{N}\big(0, \Sigma_A\big). \tag{2.31}$$

The asymptotic covariance matrix Σ_A is the unique solution of

$$\Big(AH - \frac{I}{2}\Big)\Sigma_A + \Sigma_A\Big(HA - \frac{I}{2}\Big) = A\Gamma A, \tag{2.32}$$

H and Γ being respectively the Hessian matrix of J and the covariance matrix of j, both evaluated at $u^\sharp$. Let $\mathcal{C}_H$ be the set of symmetric positive-definite matrices A, such that $AH - I/2$ is a positive-definite matrix. The next theorem characterizes the optimal choice for the gain matrix A over the set $\mathcal{C}_H$.

Theorem 2.18 *The choice $A^\sharp = H^{-1}$ for the matrix gain A in (2.30) minimizes the asymptotic covariance matrix Σ_A defined by (2.32) over the set $\mathcal{C}_H$, that is, $(\Sigma_A - \Sigma_{A^\sharp})$ is a nonnegative-definite matrix for all $A \in \mathcal{C}_H$. The expression of the minimal asymptotic covariance matrix is*

$$\Sigma_{A^\sharp} = H^{-1}\Gamma H^{-1}.$$

Proof We look for the asymptotic covariance matrix Σ_A appearing in the Lyapunov equation (2.32) in the equivalent form

$$\Sigma_A = H^{-1}\Gamma H^{-1} + \Delta_A.$$

Plugging this expression in (2.32) yields

$$\Big(AH - \frac{I}{2}\Big)\Delta_A + \Delta_A\Big(HA - \frac{I}{2}\Big) = \big(A - H^{-1}\big)\Gamma\big(A - H^{-1}\big).$$

The matrix Δ_A thus satisfies another Lyapunov equation, the right-hand side of which is a nonnegative-definite matrix whatever the choice of A. According to Proposition 2.13 and Remark 2.14, the solution Δ_A is a nonnegative-definite matrix, with $\Delta_A = 0$ if $A = H^{-1}$. We deduce that the inequality $\Sigma_A \geq H^{-1}\Gamma H^{-1}$ (in the sense of symmetric nonnegative-definite matrices) is valid for any matrix $A \in \mathcal{C}_H$, the equality being obtained for the optimal value $A^\sharp = H^{-1} \in \mathcal{C}_H$. □

Remark 2.19 The gain H^{-1} corresponds to the inverse of the Hessian matrix of J evaluated at $u^\sharp$, hence the name "Stochastic Newton Algorithm" given to (2.30) with the optimal gain choice. Note, however, that the step sizes associated with the stochastic algorithm have a length $1/k$, whereas the length is equal to 1 in the deterministic Newton algorithm. This is the reason why the convergence speeds are essentially different:

- in the deterministic case, the use of the Newton algorithm leads to a quadratic convergence speed (that is a^{2k}, with $|a| < 1$),
- whereas in the stochastic case, the convergence speed of both the scalar and the matrix gain algorithms is $a/\sqrt{k}$.

In the stochastic case, the improvement provided by using a matrix gain arises from a better multiplicative constant[9] and not from the speed $\sqrt{k}$. ◊

We give the following definition, characterizing algorithms providing the same asymptotic convergence rate as the stochastic Newton algorithm.

Definition 2.20 A stochastic gradient algorithm is *Newton-efficient* if the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ it generates has the same asymptotic convergence rate as the stochastic Newton algorithm, namely

$$\sqrt{k}\Big(\boldsymbol{U}^{(k)} - u^\sharp\Big) \xrightarrow{\mathcal{D}} \mathcal{N}\big(0, H^{-1}\Gamma H^{-1}\big).$$

According to this terminology, the iterates $\boldsymbol{U}^{(k)}$ generated by such an algorithm are asymptotically unbiased Newton-efficient estimators of $u^\sharp$.

We have seen that Newton-efficient algorithms are in some sense optimal in the stochastic gradient algorithms class. A natural question then arises. *How to implement a Newton-efficient stochastic algorithm?* The problem we have to tackle is the following: the implementation of the stochastic Newton algorithm requires the prior knowledge of the optimal gain H^{-1}, that is, the Hessian matrix of J at the solution $u^\sharp$ we are looking for! Rather than approximating H^{-1} as the algorithm runs, we now introduce an averaging method leading to a Newton-efficient algorithm.

[9] In fact a better covariance matrix.

2.5.2 Stochastic Gradient Algorithm with Averaging

In order to overcome the difficulty of implementing a Newton-efficient stochastic algorithm, in [121, 122], Polyak proposed a modification of the standard stochastic gradient method which consists in adding an averaging stage in the algorithm. More precisely, assuming that the admissible set U^{ad} is equal to the whole space $\mathbb{U}$, the standard stochastic iteration

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} - \epsilon^{(k)} \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}), \tag{2.33}$$

is replaced by

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} - \epsilon^{(k)} \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}), \tag{2.34a}$$

$$\boldsymbol{U}_{\mathrm{M}}^{(k+1)} = \frac{1}{k+1} \sum_{l=1}^{k+1} \boldsymbol{U}^{(l)}. \tag{2.34b}$$

The first stage (2.34a) is identical to (2.33), whereas the aim of the second stage (2.34b) is to compute the arithmetic mean of the iterates $\boldsymbol{U}^{(k)}$ obtained at the first stage. An equivalent recursive form for stage (2.34b) is

$$\boldsymbol{U}_{\mathrm{M}}^{(k+1)} = \boldsymbol{U}_{\mathrm{M}}^{(k)} + \frac{1}{k+1}\big(\boldsymbol{U}^{(k+1)} - \boldsymbol{U}_{\mathrm{M}}^{(k)}\big). \tag{2.34c}$$

The algorithm associated with this averaging idea is summarized as follows.

Algorithm 2.21 (Stochastic Gradient Algorithm with Averaging).

1. Select some $u^{(0)} \in U^{\mathrm{ad}}$ and choose a positive real sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$.
2. At iteration k, draw a realization $w^{(k+1)}$ of the random variable $\boldsymbol{W}$.
3. Compute the gradient of j w.r.t. u at point $(u^{(k)}, w^{(k+1)})$, and update $u^{(k+1)}$ by formula: $u^{(k+1)} = u^{(k)} - \epsilon^{(k)} \nabla_u j(u^{(k)}, w^{(k+1)})$.
4. Update $u_{\mathrm{M}}^{(k+1)}$ by formula: $u_{\mathrm{M}}^{(k+1)} = u_{\mathrm{M}}^{(k)} + \frac{1}{k+1}\big(u^{(k+1)} - u_{\mathrm{M}}^{(k)}\big)$.
5. Set $k = k+1$ and go to step 2.

As before, the value $w^{(k)}$ involved in Algorithm 2.21 is such that the sequence $(w^{(1)}, \ldots, w^{(k)})$ is a realization of a k-sample of $\boldsymbol{W}$.

Remark 2.22 Observe that $u_{\mathrm{M}}^{(k)}$ is not recycled in the algorithm, that is, the stochastic gradient is evaluated at $u^{(k)}$ and not at $u_{\mathrm{M}}^{(k)}$. This $u_{\mathrm{M}}^{(k)}$ is just an additional output of the algorithm which does not influence its dynamics. ◊

By Cesàro's lemma, the almost sure convergence of the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ implies the almost sure convergence of the averaged sequence $\{\boldsymbol{U}_{\mathrm{M}}^{(k)}\}_{k\in\mathbb{N}}$. But the salient feature of the averaged recurrence (2.34) is its asymptotic convergence speed. We use here similar assumptions as those made for Theorem 2.11, but we now suppose that the exponent γ is *strictly* smaller that 1, replacing Assumption 2.9-4 by

Assumption 2.23 The sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$ is a $\sigma(\alpha, \beta, \gamma)$-sequence, with $1/2 < \gamma < 1$.

According to Theorem 2.11, with $\gamma < 1$, the convergence speed achieved by the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ is strictly smaller than $1/\sqrt{k}$, so that the associated convergence rate is not optimal. Better convergence properties are, however, obtained regarding the averaged sequence $\{\boldsymbol{U}_{\mathrm{M}}^{(k)}\}_{k\in\mathbb{N}}$, as shown by the following theorem.

Theorem 2.24 *Under Assumptions 2.3 and 2.9, where Item 2.9-4 is replaced by Assumption 2.23, the averaged stochastic gradient algorithm is Newton-efficient:*

$$\sqrt{k}\left(\boldsymbol{U}_{\mathrm{M}}^{(k)} - u^{\sharp}\right) \xrightarrow{\mathcal{D}} \mathcal{N}\left(0, H^{-1}\Gamma H^{-1}\right).$$

For a proof, see [59, Chap. 4].

We are thus able to easily implement a Newton-efficient stochastic gradient algorithm. The averaged stochastic gradient algorithm is also referred to as the *robust approach* in Stochastic Approximation. Such a terminology is justified in Sect. 2.6.

2.5.3 Sample Average Approximation

As illustrated by Eqs. (2.33) or (2.34a), the random variables $\boldsymbol{W}^{(k)}$ are incorporated one at a time in the different versions of the stochastic gradient algorithm. Such iterative methods belong to the *Stochastic Approximation* approach (SA). There is another method, called the *Sample Average Approximation* (SAA), which makes use of all the $\boldsymbol{W}^{(k)}$ at once. As already mentioned in Sect. 2.2.1, the Sample Average Approximation method consists of replacing the expectation to be minimized by a Monte Carlo approximation. This approach is widely used in stochastic optimization for large classes of one-stage and multi-stage problems, and there is an extensive literature on Sample Average Approximation. For references on the issue of convergence[10] treated in the framework of epi-convergence, see, for example, [5, 62]. The issue of epi-convergence of the Sample Average Approximation method is also discussed in Sect. 8.4 of this book. Central Limit Theorem-like results under regularity conditions are also available ([62] and [138]), as well as results based on large deviations theory [140]. See also [141, Chap. 5] for an overview of the method, and [108] for a comparison between the Sample Average Approximation method and the Stochastic Approximation approach.

Consider Problem (2.2), and replace $J(u)$ by its Monte Carlo approximation $\boldsymbol{J}^{(k)}(u)$ obtained using a k-sample $(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(k)})$ of $\boldsymbol{W}$:

$$\boldsymbol{J}^{(k)}(u) = \frac{1}{k}\sum_{l=1}^{k} j(u, \boldsymbol{W}^{(l)}).$$

[10] Consistency in the terminology of Statistics.

The Sample Average Approximation method consists of minimizing $\boldsymbol{J}^{(k)}(u)$ for some $\omega \in \Omega$:

$$\min_{u \in U^{\mathrm{ad}}} \frac{1}{k} \sum_{l=1}^{k} j\big(u, \boldsymbol{W}^{(l)}(\omega)\big). \tag{2.35}$$

The set of minimizers of Problem (2.35) is denoted by

$$\boldsymbol{\Upsilon}^{(k)}(\omega) = \arg\min_{u \in U^{\mathrm{ad}}} \frac{1}{k} \sum_{l=1}^{k} j\big(u, \boldsymbol{W}^{(l)}(\omega)\big).$$

The properties of measurability, convergence and convergence rate of sequences $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ such that $\boldsymbol{U}^{(k)}(\omega) \in \boldsymbol{\Upsilon}^{(k)}(\omega)$ are given in [62]. Here, we just recall the main result concerning the convergence rate of such sequences [62, Theorem 4.8]. Among various technical assumptions,[11] it is assumed that

- the solution $u^\sharp$ of Problem (2.2) is unique and belongs to the interior of U^{ad},
- the function J is twice continuously differentiable with nonsingular Hessian H at $u^\sharp$,
- the sequence of random variables $\big\{\sqrt{k}\ \nabla_u \boldsymbol{J}^{(k)}(u^\sharp)\big\}_{k\in\mathbb{N}}$ converges in law to a centered gaussian distribution with covariance matrix Γ.

Then, there exists a sequence $\big\{\boldsymbol{U}^{(k)}\big\}_{k\in\mathbb{N}}$ of minimizers of (2.35) such that

$$\sqrt{k}\Big(\boldsymbol{U}^{(k)} - u^\sharp\Big) \xrightarrow{\mathcal{D}} \mathcal{N}\big(0, H^{-1}\Gamma H^{-1}\big).$$

Under mild technical assumptions, the matrix Γ is the covariance matrix of j evaluated at $u^\sharp$ (recall that $\mathbb{E}\left(\nabla_u j(u^\sharp, \boldsymbol{W})\right) = 0$) :

$$\Gamma = \mathbb{E}\left(\nabla_u j(u^\sharp, \boldsymbol{W})\big(\nabla_u j(u^\sharp, \boldsymbol{W})\big)^\top\right).$$

The asymptotic covariance matrix obtained in that case is thus equal to the optimal covariance matrix obtained when using the stochastic Newton algorithm described in Sect. 2.5.1: the sequence $\{\boldsymbol{U}^{(k)}\}_{k\in\mathbb{N}}$ generated by the Sample Average Approximation (2.35) is Newton-efficient.

2.6 Practical Considerations

In order to successfully implement a stochastic gradient algorithm, one has to keep in mind some typical difficulties that we comment upon now.

[11] See [62, Sect. 4] for further details.

2.6.1 Stopping Criterion

A first question is related to the convergence assessment of the stochastic gradient algorithm. Of course, a stopping test based on the difference norm $\|u^{(k+1)} - u^{(k)}\|$ cannot be used, since this difference is forced to zero because of the assumptions on the step sizes $\epsilon^{(k)}$. Moreover, the norm of the "descent" direction $\|\nabla_u j(u^{(k)}, w^{(k+1)})\|$ does not give any information about convergence since what is minimized is J.

However, the expectation of the random variable $\nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)})$ converges towards the true gradient $\nabla J(u^\sharp)$ at the optimum, and is accordingly usable to test the convergence. An estimation of $\nabla J(u^\sharp)$ being given by

$$\left(\sum_{l=1}^{k} \epsilon^{(l)}\right)^{-1} \left(\sum_{l=1}^{k} \epsilon^{(l)} \nabla_u j(u^{(l)}, w^{(l+1)})\right)$$

it would be possible to test whether a certain degree of convergence has been reached.

A common practice consists of fixing a given—sufficiently large—number of iterations, and to check (through plots representing the evolution of quantities related to the problem: components or norm of the variables, of the gradient…) whether convergence is achieved. This is a major difference with the deterministic case for which stopping criteria are usually available.

2.6.2 Tuning the Standard Algorithm

A fundamental issue pertains to the choice of the step sizes $\epsilon^{(k)}$. In order to satisfy the assumptions of the convergence Theorem 2.6, it seems reasonable to take $\epsilon^{(k)}$ shaped as $1/k^\gamma$, with $1/2 < \gamma \leq 1$. This is why taking a $\sigma(\alpha, \beta, \gamma)$-sequence is quite natural. The three coefficients α, β and γ, entering the choice of $\epsilon^{(k)}$ are determined according to the following guidelines.

- From Theorem 2.11, the optimal convergence rate is reached for $\gamma = 1$, leading to the well-known $1/\sqrt{k}$ rate of the Monte Carlo approximation.
- According to (2.18), the *multiplicative* coefficient α also plays a role in the asymptotic behavior. From Eq. (2.15), with $\lambda = 1/(2\alpha)$, it is easy to figure out that the covariance matrix $\alpha\Sigma$ asymptotically grows as α goes to infinity. On the other hand, using a too small value of α generates small gradient steps, which may exceedingly slow down the convergence.[12] The choice of α thus corresponds to a trade-off between stability and precision.
- Ultimately, the coefficient β makes it possible to regulate the transient behavior of the algorithm. During the first iterations, the term k^γ may be ignored w.r.t. β

[12] From Assumption 2.9-5, the condition $\alpha > 1/(2c)$ is required, c being the strong convexity modulus of J. It is easy to produce a simple problem with extremely slow convergence in the

if this is chosen large enough. The coefficient $\epsilon^{(k)}$ is approximately equal to α/β, which thus corresponds to the initial gradient step size. If α/β is too small, the transient phase may be slow. On the contrary, taking a too large ratio may lead to a numerical burst during the first iterations. Note that a first guess for the ratio α/β is given by the step size to be used by the gradient method applied to the underlying deterministic problem.

Let us illustrate the influence of parameter α with the help of a quadratic Gaussian example. The optimization problem under consideration is

$$\min_{u\in\mathbb{R}^{10}} \mathbb{E}\Big(\frac{1}{2}u^\top Au + \boldsymbol{W}^\top u\Big),$$

where A is a symmetric positive definite matrix, $\boldsymbol{W}$ being a $\mathbb{R}^{10}$-valued Gaussian random variable with expectation m and covariance matrix Γ. The solution of this problem is obviously $u^\sharp = -A^{-1}m$. The classical Monte Carlo estimator $\widehat{\boldsymbol{U}}^{(k)}$ of $u^\sharp$, namely

$$\widehat{\boldsymbol{U}}^{(k)} = -\frac{1}{k}\sum_{l=1}^{k} A^{-1}\boldsymbol{W}^{(l)}, \tag{2.36}$$

is an efficient estimator of $u^\sharp$, that is, its normalized variance reaches the Cramer-Rao lower bound (see e.g. [90] for details):

$$k\mathrm{Var}\big(\widehat{\boldsymbol{U}}^{(k)}\big) = A^{-1}\Gamma A^{-1}. \tag{2.37}$$

Using step sizes $\epsilon^{(k)} = \alpha/(k+\beta)$, the stochastic gradient iteration writes

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} - \frac{\alpha}{k+\beta}\big(A\boldsymbol{U}^{(k)} + \boldsymbol{W}^{(k+1)}\big). \tag{2.38}$$

Figure 2.1 displays four runs of the algorithm for different values of α (namely $\alpha = 0.3,\ 1.0,\ 5.0$ and 10.0), the ratio α/β being constant and equal to 0.1. For each run, we have plotted the Monte Carlo estimator ($k \mapsto \big\|\widehat{u}^{(k)} - u^\sharp\big\|$—black curve) and the stochastic gradient algorithm estimator ($k \mapsto \big\|u^{(k)} - u^\sharp\big\|$—light gray curve), where $\widehat{u}^{(k)}$ and $u^{(k)}$ correspond to realizations of the random variables $\widehat{\boldsymbol{U}}^{(k)}$ and $\boldsymbol{U}^{(k)}$ respectively. Obviously, a "small" value of $\alpha = 0.3$ (upper left-hand side plot) prevents the algorithm from converging in a reasonable time, whereas "large" values $\alpha = 5.0$ and 10.0 (lower plots) lead to excessive oscillations. In this particular example, the choice $\alpha = 1$ (upper right-hand side plot) may be considered as optimal.

(Footnote 12 continued)
case when this condition is not satisfied. For example, with $j(u, w) = (1/2)u^2$ (deterministic cost function such that $c = 1$), with $\epsilon^{(k)} = 1/(5k)$ and starting from $u^{(0)} = 1$, the solution obtained after one billion iterations is about 0.015, hence relatively far from the optimal solution $u^\sharp = 0$ (see [108] for details).

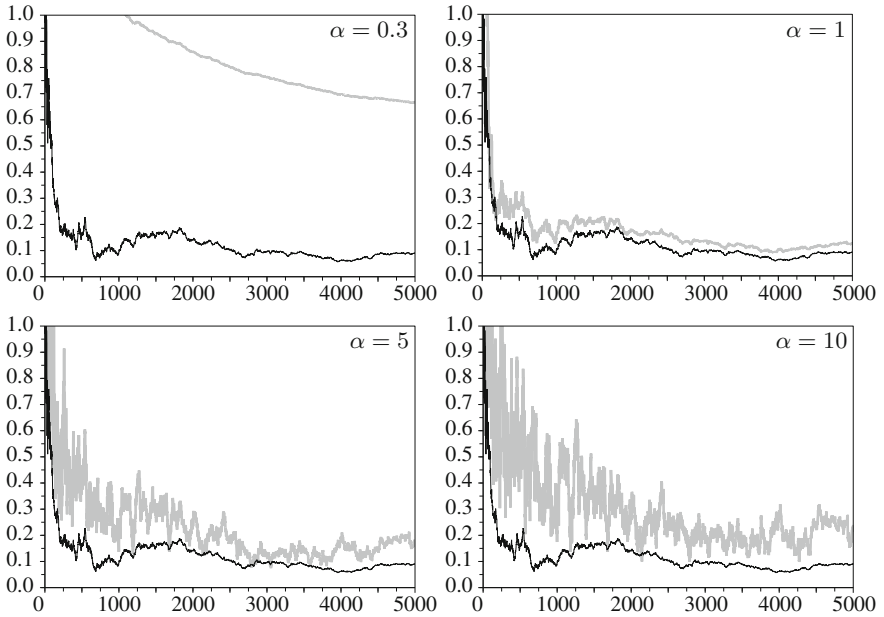

Fig. 2.1 Standard stochastic gradient runs for $\alpha = 0.3,\ 1.0,\ 5.0$ and 10.0

In order to go further into the asymptotic analysis, let us compute the covariance matrix of the iterates $\boldsymbol{U}^{(k)}$. From Eq. (2.38), denoting the identity matrix by I, we obtain that

$$\begin{aligned}\mathrm{Var}\left(\boldsymbol{U}^{(k+1)}\right) &= \mathrm{Var}\left(\left(I - \epsilon^{(k)}A\right)\boldsymbol{U}^{(k)} - \epsilon^{(k)}\boldsymbol{W}^{(k+1)}\right)\\ &= \left(I - \epsilon^{(k)}A\right)\mathrm{Var}\left(\boldsymbol{U}^{(k)}\right)\left(I - \epsilon^{(k)}A\right) + \left(\epsilon^{(k)}\right)^2\Gamma.\end{aligned}$$

The limit of the sequence of the normalized covariance matrices $k\,\mathrm{Var}\left(\boldsymbol{U}^{(k)}\right)$ induced by this relation is then compared to the Cramer-Rao bound (2.37). The lowest and greatest eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$ of these matrices are reported in Table 2.1 for different values of (α, β). We notice that the greatest eigenvalue of the Cramer-Rao bound and of the "best" covariance matrix (obtained using $\alpha = 1$) are nearly identical.

This remark enlightens a result given in [57], asserting that the greatest eigenvalue of the "optimal" covariance matrix is about $(M/c)^2$, c being the strong convexity modulus of j and M being an upper bound of the norm of the gradient of j.

As a conclusion, the implementation of the stochastic gradient algorithm is not straightforward and often requires several experiments. A common error is to consider that convergence has occurred when in fact the sequence $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$ is just badly scaled.

Table 2.1 Extreme eigenvalues of the covariance matrix for different values of (α, β)

Standard stochastic gradient algorithm	$\lambda_{\min}$	$\lambda_{\max}$
Cramer-Rao bound	**0.108**	**11.258**
$\alpha = 0.3$—$\beta = 3.0$	0.192	6170.542
$\alpha = 0.6$—$\beta = 6.0$	0.347	24.523
$\boldsymbol{\alpha = 1.0}$—$\boldsymbol{\beta = 10.0}$	0.556	**11.286**
$\alpha = 2.0$—$\beta = 20.0$	1.083	15.244
$\alpha = 5.0$—$\beta = 50.0$	2.664	32.056
$\alpha = 10.0$—$\beta = 100.0$	5.299	60.936

Remark 2.25 Many other adaptation rules have been developed in order to improve the efficiency of the stochastic gradient algorithm. For example, Chen's projection method [35]—a theoretical tool which alleviates the assumptions required for convergence in Stochastic Approximation (see [54] for further details)—also makes it possible to prevent numerical bursts in the transient phase of the algorithm. The idea is to project the iterates $\boldsymbol{U}^{(k)}$ on compact subsets of $\mathbb{U}$ forming an increasing sequence. Another approach, namely Kesten's algorithm [91], is precisely described in [55]. There, the underlying idea is to decrease the step size $\epsilon^{(k)}$ only when the directions of two consecutive gradients are opposite. More precisely, we define a (random) sequence of integers $\boldsymbol{N}_k$ by

$$\boldsymbol{N}^{(k+1)} = \boldsymbol{N}^{(k)} + \mathbf{1}_{\left\{\left\langle \nabla_u j(\boldsymbol{U}^{(k-1)}, \boldsymbol{W}^{(k)}), \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)})\right\rangle < 0\right\}},$$

$\mathbf{1}_{\Omega_0}$ being the indicator function of the set $\Omega_0 \subset \Omega$. The step size is then given by

$$\boldsymbol{\epsilon}^{(k)} = \frac{\alpha}{\left(\boldsymbol{N}^{(k)}\right)^{\gamma} + \beta}.$$

Let us mention that there exist multiplicative rules [119] for the adaptation of the step size, which allow for a faster convergence towards an approximate solution of the original problem, and that numerous references deal with stochastic algorithms using constant step sizes (see e.g. [17]). ◊

2.6.3 Robustness of the Averaged Algorithm

From a theoretical point of view, the averaged stochastic gradient is, in some sense, optimal because it has the same asymptotic convergence rate as the stochastic Newton algorithm (see Theorem 2.24). From the practical point of view, the implementation of the averaged algorithm is feasible because it does not require the knowledge of the optimal matrix gain H^{-1}. The step sizes $\epsilon^{(k)}$ form a $\sigma(\alpha, \beta, \gamma)$-sequence,

with $1/2 < k < 1$. The following considerations are relevant when choosing the parameters α, β and γ.

- The value $\gamma = 2/3$ is considered as a good choice by some authors (see [54] for further details).
- The tuning of parameters α and β is much easier than for the standard algorithm. Indeed, the problem of "too small" step sizes arising from a bad choice of α is not so critical because the term k^γ goes down more slowly towards zero. Of course, the ratio α/β must always be chosen in such a way that numerical bursts do not occur during the first iterations of the algorithm.

Remark 2.26 It seems wise not to start the averaging process from the very first iteration, because the whole transient phase of the algorithm is then taken into account in the averaged values $\boldsymbol{U}_{\mathrm{M}}^{(k)}$. It would be preferable to start the averaging process once the iterates $\boldsymbol{U}^{(k)}$ given by (2.34a) are oscillating near the convergence zone, but it is usually difficult to detect such a starting point. Another possibility is to average the stochastic gradient algorithm iterates $\boldsymbol{U}^{(k)}$ on a *sliding window*, with leads to the same asymptotic properties (see [99] for details). ◊

We now apply the averaged stochastic gradient algorithm to the example used in Sect. 2.6.2, namely

$$\begin{aligned}
\boldsymbol{U}^{(k+1)} &= \boldsymbol{U}^{(k)} - \frac{\alpha}{k^\gamma + \beta}\left(A\boldsymbol{U}^{(k)} + \boldsymbol{W}^{(k+1)}\right),\\
\boldsymbol{U}_{\mathrm{M}}^{(k+1)} &= \frac{1}{k+1}\sum_{l=1}^{k+1} \boldsymbol{U}^{(l)}.
\end{aligned}$$

We use the same values of α and β as for the standard stochastic algorithm, γ being now equal to $2/3$. The four runs of the averaged algorithm are plotted in Fig. 2.2. For each run, we have again plotted the Monte Carlo estimator given by (2.36) ($k \mapsto \|\widehat{u}^{\,(k)} - u^\sharp\|$—black curve), the stochastic gradient algorithm estimator ($k \mapsto \|u^{(k)} - u^\sharp\|$—light gray curve), and finally the averaged stochastic gradient algorithm estimator ($k \mapsto \left\|u_{\mathrm{M}}^{(k)} - u^\sharp\right\|$—dark gray curve). The changes of parameter α (from 0.3 to 10.0) affect the behavior of the stochastic gradient algorithm estimator, the oscillations of which increase with α. Nevertheless, the behavior of the averaged stochastic gradient algorithm estimator remains remarkably stable, hence the term "robust" given to the averaged algorithm.

It is again possible to iteratively compute the covariance matrices of the iterates $\boldsymbol{U}_{\mathrm{M}}^{(k)}$. The lowest and greatest eigenvalues of these matrices are given in Table 2.2 for the different values of α. We observe that the full spectrum of the Cramer-Rao bound is obtained whatever the value of α.

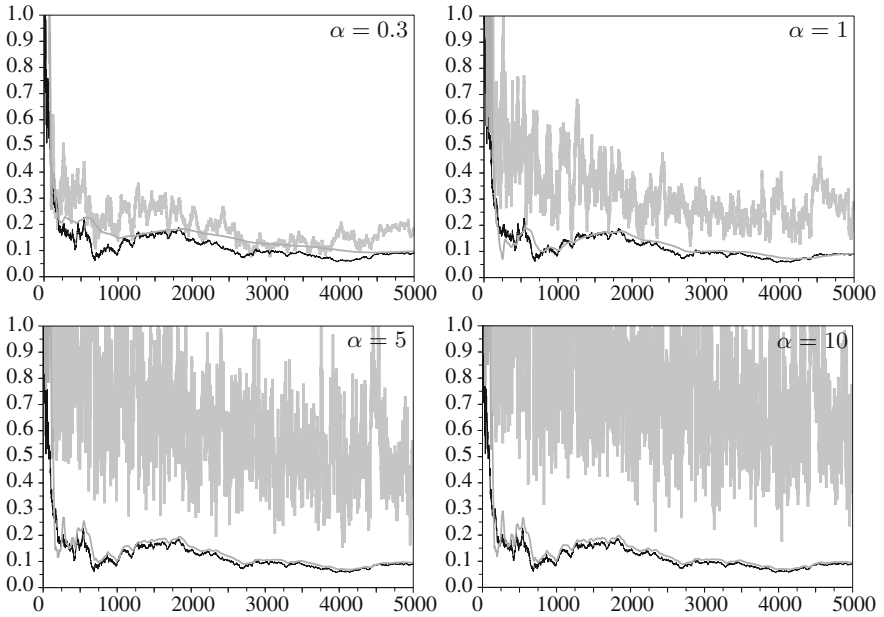

Fig. 2.2 Averaged stochastic gradient runs for $\alpha = 0.3,\ 1.0,\ 5.0$ and 10.0

Table 2.2 Extreme eigenvalues of the covariance matrix for different values of (α, β)

Averaged stochastic gradient algorithm	$\lambda_{\min}$	$\lambda_{\max}$
Cramer-Rao bound	**0.108**	**11.258**
$\alpha = 0.3$—$\beta = 3.0$	0.108	11.360
$\alpha = 0.5$—$\beta = 5.0$	0.108	11.318
$\alpha = 1.0$—$\beta = 10.0$	0.108	11.288
$\alpha = 2.0$—$\beta = 20.0$	0.108	11.273
$\alpha = 5.0$—$\beta = 50.0$	0.108	11.264
$\alpha = 10.0$—$\beta = 100.0$	0.108	11.262

2.7 Conclusion

In this chapter, we have tried to give a broad (of course non exhaustive) overview of the stochastic gradient method. After recalling some classical results from Stochastic Approximation, we have presented an algorithm based on both the Stochastic Gradient Method and on the Auxiliary Problem Principle, for which we provided a detailed convergence analysis. We then presented some issues related to the efficiency of the stochastic gradient algorithm. Finally, we have made some practical considerations about the algorithm implementation. Note that this domain is still very active, as demonstrated by the recent paper [162] providing new adaptive step length

schemes in order to improve the performance of stochastic gradient algorithms, and by the paper [108] comparing the Sample Average Approximation method with a properly modified Stochastic Approximation approach. About the last paper, it is interesting to remark the strong connections between the Mirror Descent Stochastic Approximation method and the Auxiliary Problem Principle. Although restricted to the computation of open-loop solutions,[13] the stochastic gradient method is a basic component of stochastic optimization which can be embedded in many dynamic situations, when some control variables have to be decided upon once and for all or some static parameters have to be tuned. It is the case for two-stage stochastic optimization problems, for which the first time step decisions are open-loop decisions. It is also the case for multistage stochastic optimization problems when it is possible to restrict the admissible feedback laws to a particular class of functions which can be characterized in terms of a finite number of parameters, e.g., (s, S)-policies, impulse control, etc. See [148], and also [145] for a more recent application.

Throughout this book, in addition to the challenge of dealing with expectations (which was the main purpose of this chapter), we will deal with the additional difficulty related to the issue of information, that is, the measurability constraints.

2.8 Appendix

This last section is devoted to the proof of the main convergence Theorem 2.17. The proof is based on two results, namely the Robbins-Siegmund theorem and a technical lemma, that are beforehand recalled.

2.8.1 Robbins-Siegmund Theorem

The following theorem is one of the keys to Stochastic Approximation.

Theorem 2.27 *Let* $\{\boldsymbol{\Lambda}^{(k)}\}_{k\in\mathbb{N}}$, $\{\boldsymbol{\alpha}^{(k)}\}_{k\in\mathbb{N}}$, $\{\boldsymbol{\beta}^{(k)}\}_{k\in\mathbb{N}}$ *and* $\{\boldsymbol{\eta}^{(k)}\}_{k\in\mathbb{N}}$ *be four positive sequences of real-valued random variables adapted to the filtration* $\{\mathcal{F}^{(k)}\}_{k\in\mathbb{N}}$. *Assume that*

$$\mathbb{E}\big(\boldsymbol{\Lambda}^{(k+1)} \,\big|\, \mathcal{F}^{(k)}\big) \leq \big(1+\boldsymbol{\alpha}^{(k)}\big)\boldsymbol{\Lambda}^{(k)} + \boldsymbol{\beta}^{(k)} - \boldsymbol{\eta}^{(k)}, \ \ \forall k \in \mathbb{N},$$

and that

$$\sum_{k\in\mathbb{N}} \boldsymbol{\alpha}^{(k)} < +\infty \ \ \textit{and} \ \ \sum_{k\in\mathbb{N}} \boldsymbol{\beta}^{(k)} < +\infty, \ \ \mathbb{P}\textit{-a.s.}.$$

[13] There however exist extensions of the stochastic gradient method to closed-loop optimization problem: see [14] for further details.

Then, the sequence $\{\boldsymbol{\Lambda}^{(k)}\}_{k\in\mathbb{N}}$ *almost surely converges to a finite*[14] *random variable* $\boldsymbol{\Lambda}^{\infty}$, *and* $\sum_{k\in\mathbb{N}} \boldsymbol{\eta}^{(k)} < +\infty$, $\mathbb{P}$-*a.s.*.

A proof can be found e.g. in [60, Theorem 1.3.12].

2.8.2 A Technical Lemma

The following lemma is also used in order to prove the convergence of the stochastic APP algorithm.

Lemma 2.28 *Let* J *be a real-valued function defined on a Hilbert space* $\mathbb{U}$. *We assume that* J *is Lipschitz continuous with constant* L. *Let* $\{u^{(k)}\}_{k\in\mathbb{N}}$ *be a sequence of elements of* $\mathbb{U}$ *and let* $\{\epsilon^{(k)}\}_{k\in\mathbb{N}}$ *be a sequence of positive real numbers such that*

(a) $\displaystyle\sum_{k\in\mathbb{N}} \epsilon^{(k)} = +\infty,$

(b) $\exists\, \mu \in \mathbb{R},\ \displaystyle\sum_{k\in\mathbb{N}} \epsilon^{(k)} \left|J(u^{(k)}) - \mu\right| < +\infty,$

(c) $\exists\, \delta > 0,\ \forall k \in \mathbb{N},\ \left\|u^{(k+1)} - u^{(k)}\right\| \leq \delta\epsilon^{(k)}.$

Then the sequence $\left\{J(u^{(k)})\right\}_{k\in\mathbb{N}}$ *converges to* μ.

Proof Let α be a given positive real number. We define the subset N_α of $\mathbb{N}$ and its complementary N_α^{c} as follows:

$$N_\alpha = \left\{k \in \mathbb{N},\ \left|J(u^{(k)}) - \mu\right| \leq \alpha\right\} \quad \text{and} \quad N_\alpha^{\mathrm{c}} = \mathbb{N} \setminus N_\alpha.$$

From the definition of N_α^{c}, we have that

$$\sum_{k\in N_\alpha^{\mathrm{c}}} \epsilon^{(k)} \left|J(u^{(k)}) - \mu\right| \geq \alpha \sum_{k\in N_\alpha^{\mathrm{c}}} \epsilon^{(k)},$$

and Property (b) implies that

$$\sum_{k\in N_\alpha^{\mathrm{c}}} \epsilon^{(k)} \left|J(u^{(k)}) - \mu\right| \leq \sum_{k\in\mathbb{N}} \epsilon^{(k)} \left|J(u^{(k)}) - \mu\right| < +\infty.$$

We thus deduce that the series $\sum_{k\in N_\alpha^{\mathrm{c}}} \epsilon^{(k)}$ converges, that is,

$$\forall \beta > 0,\ \exists n_\beta \in \mathbb{N},\ \sum_{k\in N_\alpha^{\mathrm{c}},\, k\geq n_\beta} \epsilon^{(k)} \leq \beta. \tag{2.39}$$

[14] A random variable $\boldsymbol{X}$ is finite if $\mathbb{P}\big(\left\{\omega \in \Omega \ \middle|\ \boldsymbol{X}(\omega) = +\infty\right\}\big) = 0$.

Then, from (2.39) and Property (a), we obtain that N_α is not a finite set.

For each $\epsilon > 0$, we choose $\alpha = \epsilon/2$ and $\beta = \epsilon/(2L\delta)$. Let n_β be the integer defined by (2.39). For any $k \geq n_\beta$,

- either $k \in N_\alpha$ and, we have, by definition

$$\left|J(u^{(k)}) - \mu\right| \leq \alpha < \epsilon,$$

- or $k \in N_\alpha^c$; then let m be the smallest element of N_α such that $m > k$ (such an element exists because N_α is not a finite set); using the Lipschitz assumption on J and Property (c), we obtain

$$\begin{aligned}\left|J(u^{(k)}) - \mu\right| &\leq \left|J(u^{(k)}) - J(u^{(m)})\right| + \left|J(u^{(m)}) - \mu\right| \leq L\left\|u^{(k)} - u^{(m)}\right\| + \alpha \\ &\leq L\delta\left(\sum_{l=k}^{m-1} \epsilon^{(l)}\right) + \alpha \leq L\delta\left(\sum_{l \geq n_\beta, l \in N_\alpha^c} \epsilon^{(l)}\right) + \alpha \leq \epsilon,\end{aligned}$$

hence the result. □

2.8.3 *Proof of Theorem 2.17*

Here we give the complete proof of the main convergence theorem.

Proof The proof of the first statement is based on classical theorems in the field of convex optimization (see Theorem A.8). The existence of a random variable $\boldsymbol{U}^{(k+1)}$ solution of Problem (2.24) is a consequence of the fact that the criterion to be minimized in (2.24) is a normal integrand, so that the arg min is closed-valued and measurable, and thus admits measurable selections (see [135, Theorem 14.37] for further details). The solution $\boldsymbol{U}^{(k+1)}$ is unique because K is strongly convex.

The proof of the last two statements involves four steps.

Select a Lyapunov function Λ. Let $u^\sharp \in U^\sharp$ be a solution of (2.22). We consider the so-called Lyapunov function $\Lambda : \mathbb{U} \to \mathbb{R}$, defined by

$$\Lambda(u) = K(u^\sharp) - K(u) - \left\langle \nabla K(u)\,, u^\sharp - u \right\rangle.$$

From the strong convexity of K, we have that

$$\frac{b}{2}\left\|u - u^\sharp\right\|^2 \leq \Lambda(u). \tag{2.40}$$

The Lyapunov function Λ is thus bounded from below and coercive.

Bound from above the variation of Λ. We consider the difference

$$\Delta^{(k)} = \Lambda(u^{(k+1)}) - \Lambda(u^{(k)}),$$

$\{u^{(k)}\}_{k\in\mathbb{N}}$ being the sequence of solutions generated by Algorithm 2.15 for a realization $(w^{(1)}, \dots, w^{(k)}, \dots)$ of the infinite-dimensional sample of $\boldsymbol{W}$:

$$\Delta^{(k)} = \underbrace{K(u^{(k)}) - K(u^{(k+1)}) - \left\langle \nabla K(u^{(k)}) \, , u^{(k)} - u^{(k+1)} \right\rangle}_{T_1} + \underbrace{\left\langle \nabla K(u^{(k)}) - \nabla K(u^{(k+1)}) \, , u^{\sharp} - u^{(k+1)} \right\rangle}_{T_2}.$$

- From the convexity of K, we have that

$$T_1 \leq 0.$$

- Let $r^{(k)} = \nabla_u j(u^{(k)}, w^{(k+1)})$. The optimality condition of Problem (2.23) writes

$$\left\langle \nabla K(u^{(k+1)}) + \epsilon^{(k)} r^{(k)} - \nabla K(u^{(k)}) \, , u - u^{(k+1)} \right\rangle \geq 0, \quad \forall u \in U^{\mathrm{ad}}. \tag{2.41}$$

  Evaluating (2.41) at $u = u^{\sharp}$ leads to

$$\begin{aligned} T_2 &\leq \epsilon^{(k)} \left\langle r^{(k)} \, , u^{\sharp} - u^{(k+1)} \right\rangle \\ &\leq \epsilon^{(k)} \underbrace{\left\langle r^{(k)} \, , u^{\sharp} - u^{(k)} \right\rangle}_{T_3} + \epsilon^{(k)} \underbrace{\left\langle r^{(k)} \, , u^{(k)} - u^{(k+1)} \right\rangle}_{T_4}. \end{aligned}$$

  – From the convexity of $j(\cdot, w^{(k+1)})$, we have that

$$T_3 \leq j(u^{\sharp}, w^{(k+1)}) - j(u^{(k)}, w^{(k+1)}).$$

  – The evaluation of (2.41) at $u = u^{(k)}$ and the strong monotonicity of ∇K imply that

$$b \left\| u^{(k+1)} - u^{(k)} \right\|^2 \leq \epsilon^{(k)} \left\langle r^{(k)} \, , u^{(k)} - u^{(k+1)} \right\rangle.$$

  Using the Schwartz inequality, we obtain

$$\left\| u^{(k+1)} - u^{(k)} \right\| \leq \frac{\epsilon^{(k)}}{b} \left\| r^{(k)} \right\|. \tag{2.42}$$

  Applying also the Schwartz inequality to the term T_4 and using (2.42) yield

$$T_4 \leq \frac{\epsilon^{(k)}}{b} \left\| r^{(k)} \right\|^2.$$

  An equivalent form for the LBG assumption is that there exist positive constants c_3 and c_4 such that $\left\| r^{(k)} \right\| \leq c_3 \left\| u^{(k)} - u^{\sharp} \right\| + c_4$. Taking the square of

the last inequality, using $(a+b)^2 \leq 2(a^2+b^2)$ as well as (2.40), we obtain that

$$\exists \alpha > 0,\ \exists \beta > 0,\ \forall k \in \mathbb{N},\ \left\| r^{(k)} \right\|^2 \leq \alpha \Lambda(u^{(k)}) + \beta,$$

and, consequently,

$$T_4 \leq \frac{\epsilon^{(k)}}{b} \left(\alpha \Lambda(u^{(k)}) + \beta \right).$$

Collecting the upper bounds obtained for T_1, T_3 and T_4, we deduce that

$$\Delta^{(k)} \leq \epsilon^{(k)} \left(j(u^\sharp, w^{(k+1)}) - j(u^{(k)}, w^{(k+1)}) \right) + \frac{\left(\epsilon^{(k)}\right)^2}{b} \left(\alpha \Lambda(u^{(k)}) + \beta \right).$$

Consider this inequality in terms of random variables. Taking the conditional expectation w.r.t. the σ-field $\mathcal{F}^{(k)}$ generated by $(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(k)})$ on both sides, recalling that $\boldsymbol{W}^{(k+1)}$ is independent of the previous $\boldsymbol{W}^{(l)}$ and that $\boldsymbol{U}^{(k)}$ is $\mathcal{F}^{(k)}$-measurable, we obtain that

$$\begin{aligned} \mathbb{E}\left(\Lambda(\boldsymbol{U}^{(k+1)}) - \Lambda(\boldsymbol{U}^{(k)}) \mid \mathcal{F}^{(k)} \right) \leq \alpha^{(k)} \mathbb{E}\left(\Lambda(\boldsymbol{U}^{(k)}) \mid \mathcal{F}^{(k)} \right) + \beta^{(k)} \\ + \epsilon^{(k)} \left(J(u^\sharp) - J(\boldsymbol{U}^{(k)}) \right), \end{aligned} \tag{2.43}$$

$\alpha^{(k)} = (\alpha/b)(\epsilon^{(k)})^2$ and $\beta^{(k)} = (\beta/b)(\epsilon^{(k)})^2$ being the terms of two convergent series. Thanks to the optimality of $u^\sharp$, we have that $J(u^\sharp) - J(\boldsymbol{U}^{(k)}) \leq 0$.

Convergence. A straightforward application of the Robbins-Siegmund Theorem 2.27 shows that the sequence $\left\{ \Lambda(\boldsymbol{U}^{(k)}) \right\}_{k \in \mathbb{N}}$ almost surely converges to a finite random variable $\boldsymbol{\Lambda}^\infty$, and that

$$\sum_{k=0}^{+\infty} \epsilon^{(k)} \left(J(\boldsymbol{U}^{(k)}) - J(u^\sharp) \right) < +\infty,\ \mathbb{P}\text{-a.s.}. \tag{2.44}$$

Sequence Limit. As proved in the previous step, the sequence $\left\{ \Lambda(\boldsymbol{U}^{(k)}) \right\}_{k \in \mathbb{N}}$ almost surely converges to a finite random variable, and hence is almost surely bounded. According to (2.40) and the LBG assumption, we deduce that both sequences $\{\boldsymbol{U}^{(k)}\}_{k \in \mathbb{N}}$ and $\left\{ \nabla_u j(\boldsymbol{U}^{(k)}, \boldsymbol{W}^{(k+1)}) \right\}_{k \in \mathbb{N}}$ are almost surely bounded. Thanks to (2.42), the same holds true for the sequence $\left\{ \|\boldsymbol{U}^{(k+1)} - \boldsymbol{U}^{(k)}\| / \epsilon^{(k)} \right\}_{k \in \mathbb{N}}$. This last fact together with (2.44) make it possible to use Lemma 2.28 to claim that the sequence $\{J(\boldsymbol{U}^{(k)})\}_{k \in \mathbb{N}}$ almost surely converges to $J(u^\sharp)$.

Let Ω_0 denote the subset of Ω such that $\left\{ \Lambda(\boldsymbol{U}^{(k)}) \right\}_{k \in \mathbb{N}}$ is not bounded, and let Ω_1 denote the subset of Ω for which (2.44) does not hold: $\mathbb{P}(\Omega_0 \cup \Omega_1) = 0$. Pick some $\omega \notin \Omega_0 \cup \Omega_1$. The sequence of realizations $\{u^{(k)}\}_{k \in \mathbb{N}}$ of $\{\boldsymbol{U}^{(k)}\}_{k \in \mathbb{N}}$ associated with ω is bounded, and each $u^{(k)}$ belongs to the closed subset U^{ad}. By a

compactness argument,[15] there exists a convergent subsequence $\{u^{(\Phi(k))}\}_{k\in\mathbb{N}}$ (note that the subsequence itself depends on ω); let $\bar{u}$ be the limit of this subsequence. By the lower semi-continuity of function J, we have that

$$J(\bar{u}) \leq \liminf_{k\to+\infty} J(u^{(\Phi(k))}) = J(u^\sharp).$$

We thus deduce that $\bar{u} \in U^\sharp$.

We ultimately consider the case when J is strongly convex with modulus a. Then Problem (2.22) has a unique solution $u^\sharp$. Thanks to the optimality condition (2.20), the strong convexity property of J writes

$$\begin{aligned} J(\boldsymbol{U}^{(k)}) - J(u^\sharp) &\geq \left\langle \nabla J(u^\sharp)\,,\, \boldsymbol{U}^{(k)} - u^\sharp \right\rangle + \frac{a}{2}\left\| \boldsymbol{U}^{(k)} - u^\sharp \right\|^2 \\ &\geq \frac{a}{2}\left\| \boldsymbol{U}^{(k)} - u^\sharp \right\|^2. \end{aligned}$$

Since $J(\boldsymbol{U}^{(k)})$ converges almost surely to $J(u^\sharp)$, we deduce that $\left\| \boldsymbol{U}^{(k)} - u^\sharp \right\|$ almost surely converges to zero. The proof is complete. □

[15] A subset of $\mathbb{U}$ is compact if it is closed and bounded, provided that $\mathbb{U}$ is a finite-dimensional space. If $\mathbb{U}$ is an infinite-dimensional Hilbert space, such a property remains true only in the weak topology, and the lower semi-continuity property of J is preserved in that topology because J is convex. See [64] for further details.

http://www.springer.com/978-3-319-18137-0

Stochastic Multi-Stage Optimization
At the Crossroads between Discrete Time Stochastic Control and Stochastic Programming
Carpentier, P.; Chancelier, J.-P.; Cohen, G.; De Lara, M.
2015, XVII, 362 p. 45 illus., 14 illus. in color., Hardcover
ISBN: 978-3-319-18137-0