Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. James C. Spall

ISBN: 0-471-33052-3

CHAPTER 7

SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION

Chapters 5 and 6 discussed the use of stochastic approximation (SA) for problems of minimizing a loss function $L(\boldsymbol{\theta})$. Chapter 5 considered the case where direct unbiased measurements of the gradient $\boldsymbol{g}(\boldsymbol{\theta})$ are available. Chapter 6, on the other hand, introduced the notion of gradient-free SA, where optimization is carried out with only noisy measurements of the loss function. This is motivated by the many problems where direct measurements of the gradient are not available. This chapter explores a method—simultaneous perturbation stochastic approximation (SPSA)—applicable in both the stochastic gradient and gradient-free settings. SPSA typically offers significant efficiency gains in problems with a large number of variables to be optimized.

Section 7.1 is a brief introduction. Section 7.2 describes the basic SPSA algorithm and contrasts it with the finite-difference SA (FDSA) algorithm of Chapter 6. Sections 7.3 and 7.4 discuss some of the theory associated with the convergence and asymptotic normality of SPSA, much like the theory for the root-finding SA and FDSA algorithms considered earlier. The asymptotic normality of the iterate is used to draw some powerful conclusions about the relative efficiency of SPSA and FDSA. These efficiency results provide the main rationale for using SPSA instead of FDSA in practical applications. Section 7.5 presents a step-by-step guide to implementation that is aimed at helping the reader code the algorithm for his or her specific application. This section also summarizes some additional implementation aspects regarding the choice of algorithm gain sequences. Section 7.6 presents some numerical results, including results that illustrate the theoretical efficiency conclusions in Section 7.4. Section 7.7 briefly discusses some extensions to the basic SPSA algorithm, including modifications to perform discrete optimization (discrete $\boldsymbol{\theta}$), global optimization, and constrained optimization. Section 7.8 summarizes some relatively recent results on a second-order (adaptive) version of SPSA that emulates for stochastic problems the Newton–Raphson algorithm of deterministic optimization. This adaptive SPSA approach applies in either the "standard" gradient-free setting of Chapter 6, where only (noisy) loss function measurements are available, or in the stochastic gradient (Robbins–Monro) setting of Chapter 5, where direct unbiased gradient measurements are available.

Section 7.9 offers some concluding remarks and Section 7.10 is an appendix containing conditions for asymptotic normality of the iterate.

7.1 BACKGROUND

Chapters 5 and 6 demonstrated that stochastic approximation applies to a large number of optimization problems where only noisy measurements of a criterion are available. In the stochastic gradient framework (Chapter 5), a direct unbiased measurement of the gradient $\boldsymbol{g}(\boldsymbol{\theta}) = \partial L/\partial \boldsymbol{\theta}$ is used in the SA algorithm. Chapter 6, on the other hand, works with gradient *approximations* built from (noisy) measurements of L. Such methods are useful when it is very costly or impossible to directly measure the gradient $\boldsymbol{g}(\boldsymbol{\theta})$ at different values of $\boldsymbol{\theta}$. Continuing in the spirit of Chapter 6, where only noisy loss measurements are available, this chapter discusses the simultaneous perturbation stochastic approximation (SPSA) algorithm for stochastic optimization of multivariate systems. Relative to the finite-difference-based methods of Chapter 6, the principal benefit of SPSA is a reduction in the number of loss measurements required to achieve a given level of accuracy in the optimization process.

The central focus with SPSA is the stochastic setting where only measurements of the loss function are available (i.e., no gradient information). That is, the algorithm is based on loss measurements $y(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \varepsilon$ at various values of $\boldsymbol{\theta}$; equivalently, as in Chapter 5, one can write $y(\boldsymbol{\theta}) = Q(\boldsymbol{\theta}, V)$, where Q represents some observed cost as a function of the chosen $\boldsymbol{\theta}$ and random effects V (so $\varepsilon = \varepsilon(\boldsymbol{\theta}) = Q(\boldsymbol{\theta}, V) - L(\boldsymbol{\theta})$). More recent results, however, show that the SPSA idea can be extended in a relatively simple way to the stochastic gradient setting, providing an efficient means for building an asymptotically optimal stochastic analogue of the Newton–Raphson algorithm (Section 7.8). Hence, the SPSA principles can be used in either the stochastic gradient (Chapter 5) or gradient-free (Chapter 6) settings. As we have seen, the interest in both the stochastic gradient and gradient-free SA algorithms has been motivated by problems such as adaptive control, model parameter estimation, and simulation-based optimization.

We saw in Chapter 6 that finite-difference SA (FDSA) exhibits convergence properties similar to the stochastic gradient-based stochastic algorithms while requiring only loss function measurements. SPSA shares this property with FDSA. The asymptotic normality of SPSA and FDSA can be used to draw fundamental conclusions on the relative large-sample efficiency of the approaches.

The SPSA Web site (*www.jhuapl.edu/SPSA*; also accessible through this book's Web site) includes references describing applications in areas such as queuing systems, industrial quality improvement, aircraft design, pattern recognition, simulation-based optimization (with applications, e.g., to air traffic management and military planning), bioprocess control, neural network training,

chemical process control, fault detection, human-machine interaction, sensor placement and configuration, and vehicle traffic management.

7.2 FORM AND MOTIVATION FOR STANDARD SPSA ALGORITHM

7.2.1 Basic Algorithm

This section is devoted to the "basic" SPSA algorithm, which applies in the gradient-free setting of FDSA. (Section 7.8 considers the adaptive version of SPSA that applies in either the gradient-free or the stochastic gradient-based setting.) As motivated above, we now assume that no direct measurements of $\boldsymbol{g}(\boldsymbol{\theta})$ are available. Following the mold of Chapter 6, the basic unconstrained SPSA algorithm is in the general recursive SA form:

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k \hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k), \tag{7.1}$$

where $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ is the simultaneous perturbation estimate of the gradient $\boldsymbol{g}(\boldsymbol{\theta}) = \partial L/\partial \boldsymbol{\theta}$ at the iterate $\hat{\boldsymbol{\theta}}_k$ based on the measurements of the loss function and a_k is a nonnegative scalar gain coefficient. (The constrained case is briefly discussed in Section 7.7.)

The essential part of (7.1) is the gradient approximation $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$. Recall that with FDSA, this gradient approximation is formed by perturbing the components of $\hat{\boldsymbol{\theta}}_k$ one at a time and collecting a loss measurement $y(\cdot)$ at each of the perturbations (in practice, the loss measurements are sometimes noise-free, a la $y(\cdot) = L(\cdot)$). This requires $2p$ loss measurements for a two-sided FD approximation. In contrast, with simultaneous perturbation, all elements of $\hat{\boldsymbol{\theta}}_k$ are randomly perturbed together to obtain two loss measurements $y(\cdot)$. For the two-sided SP gradient approximation, this leads to

$$\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k) = \begin{bmatrix} \dfrac{y(\hat{\boldsymbol{\theta}}_k + c_k\Delta_k) - y(\hat{\boldsymbol{\theta}}_k - c_k\Delta_k)}{2c_k\Delta_{k1}} \\ \vdots \\ \dfrac{y(\hat{\boldsymbol{\theta}}_k + c_k\Delta_k) - y(\hat{\boldsymbol{\theta}}_k - c_k\Delta_k)}{2c_k\Delta_{kp}} \end{bmatrix}$$

$$= \frac{y(\hat{\boldsymbol{\theta}}_k + c_k\Delta_k) - y(\hat{\boldsymbol{\theta}}_k - c_k\Delta_k)}{2c_k} \left[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \ldots, \Delta_{kp}^{-1}\right]^T, \tag{7.2}$$

where the mean-zero p-dimensional random perturbation vector, $\Delta_k = [\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp}]^T$, has a user-specified distribution satisfying conditions discussed in Sections 7.3 and 7.4 and c_k is a positive scalar. Because the numerator is the same in all p components of $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$, the number of loss measurements needed to estimate the gradient in SPSA is *two*, regardless of the dimension p. Recall that Section 6.8 discussed a similar random directions gradient approximation.

While the number of loss function measurements $y(\cdot)$ needed in each iteration of FDSA grows with p, the number in SPSA is fixed. This measurement savings per iteration, of course, provides only the *potential* for SPSA to achieve large savings (over FDSA) in the total number of measurements required to estimate $\boldsymbol{\theta}$ when p is large. This potential is realized if the number of iterations required for effective convergence to an optimum $\boldsymbol{\theta}^*$ does not increase in a way to cancel the measurement savings per gradient approximation at each iteration. We would expect this potential to be realized if, roughly speaking, the FD and SP gradient approximations acted the same in some statistical sense *relative to their use in the basic optimization recursion* (which is the same basic form in both FDSA and SPSA).

It is clear that the SP approximation above will *not* act the same as the FD approximation as an estimate of the gradient per se. The FD approximation will generally be superior in that sense. However, the interest is not in the gradient per se. Rather, the interest is in how the approximations operate *when considered in optimization* over multiple iterations with a changing point of evaluation $\boldsymbol{\theta}$. Section 7.4 discusses the efficiency issue further, establishing the fundamental result:

> Under reasonably general conditions (see Section 7.4), the SPSA and FDSA recursions achieve the same level of statistical accuracy for a given number of iterations even though SPSA uses only $1/p$ times the number of function evaluations of FDSA (since each gradient approximation uses only $1/p$ the number of function evaluations).

7.2.2 Relationship of Gradient Estimate to True Gradient

The informal rationale for the strange-looking gradient approximation in (7.2) is quite simple. Consider the mth element of the approximation. Let us sketch how this element is an "almost unbiased" estimator of the mth element of the true gradient. The formal results on the bias to follow and the convergence and asymptotic normality in Sections 7.3 and 7.4 make this more rigorous. Suppose that the measurement noise $\varepsilon(\boldsymbol{\theta})$ has mean zero and that L is several times differentiable at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_k$. Then, using a simple first-order Taylor expansion,

$$
\begin{aligned}
E\left[\hat{g}_{km}(\hat{\boldsymbol{\theta}}_k)\,|\,\hat{\boldsymbol{\theta}}_k\right] &= E\left[\frac{y(\hat{\boldsymbol{\theta}}_k + c_k\Delta_k) - y(\hat{\boldsymbol{\theta}}_k - c_k\Delta_k)}{2c_k\Delta_{km}}\,\middle|\,\hat{\boldsymbol{\theta}}_k\right] \quad \text{(from (7.2))} \\
&= E\left[\frac{L(\hat{\boldsymbol{\theta}}_k + c_k\Delta_k) - L(\hat{\boldsymbol{\theta}}_k - c_k\Delta_k)}{2c_k\Delta_{km}}\,\middle|\,\hat{\boldsymbol{\theta}}_k\right] \quad \text{(noise terms disappear)} \\
&\approx E\left[\frac{L(\hat{\boldsymbol{\theta}}_k) + c_k\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k)^T\Delta_k - [L(\hat{\boldsymbol{\theta}}_k) - c_k\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k)^T\Delta_k]}{2c_k\Delta_{km}}\,\middle|\,\hat{\boldsymbol{\theta}}_k\right] \quad \begin{array}{l}\text{(first-order}\\ \text{expansions of}\\ L(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k))\end{array} \\
&= E\left[\frac{2c_k\sum_{i=1}^{p} L_i'(\hat{\boldsymbol{\theta}}_k)\Delta_{ki}}{2c_k\Delta_{km}}\,\middle|\,\hat{\boldsymbol{\theta}}_k\right] \quad \text{(cancel } L \text{ terms)} \\
&= L_m'(\hat{\boldsymbol{\theta}}_k) + \sum_{i\neq m} L_i'(\hat{\boldsymbol{\theta}}_k)E\left(\frac{\Delta_{ki}}{\Delta_{km}}\right) \quad \text{(rewrite of above)}, \qquad (7.3)
\end{aligned}
$$

where $L_i'(\cdot)$ denotes the ith component of $\boldsymbol{g}(\cdot)$. Assume that Δ_{ki} has mean zero and is independent of Δ_{km} for all $i \neq m$. Then, to ensure that $E(\Delta_{ki}/\Delta_{km})$ in the last line of (7.3) represents a valid expectation, it is necessary that the inverse moment $E(|1/\Delta_{km}|)$ be finite. (See Exercise 7.2, recalling the formal definition of expectation in Section C.1 of Appendix C.) The algorithm convergence results in Sections 7.3 and 7.4 use a more stringent version of this inverse moment condition, but the basic idea is the same; see the comments after conditions B.1″–B.6″ in Section 7.3 regarding the implication of this inverse moment condition. Given the above, $E(\Delta_{ki}/\Delta_{km}) = 0$ for all $i \neq m$.

Hence, the term after the "+" sign in the bottom line of (7.3) disappears, indicating that

$$
E[\hat{g}_{km}(\hat{\boldsymbol{\theta}}_k)\,|\,\hat{\boldsymbol{\theta}}_k] \approx L_m'(\hat{\boldsymbol{\theta}}_k), \qquad (7.4)
$$

as desired. The precise meaning of the "≈" in (7.4) is given below, but one can see from (7.3) that the bias in the gradient approximation (i.e., the difference between an "=" and the indicated "≈") is due to the higher-order terms in the expansion of L appearing on the third line. As with FDSA, the resulting bias is $O(c_k^2)$, although the exact form of the bias differs from FDSA.

Figure 7.1 provides a pictorial representation of SPSA for the case where, for all k and i, the Δ_{ki} are generated by a symmetric Bernoulli ± 1 distribution (i.e., with probability 1/2, there is one of two possible outcomes: $\Delta_{ki} = 1$ or $\Delta_{ki} = -1$). As discussed in Sections 7.3 and 7.7, the symmetric Bernoulli distribution is an important special case among the distributions that satisfy the conditions for Δ_k (symmetric, mean zero, finite variance, and finite inverse moments).

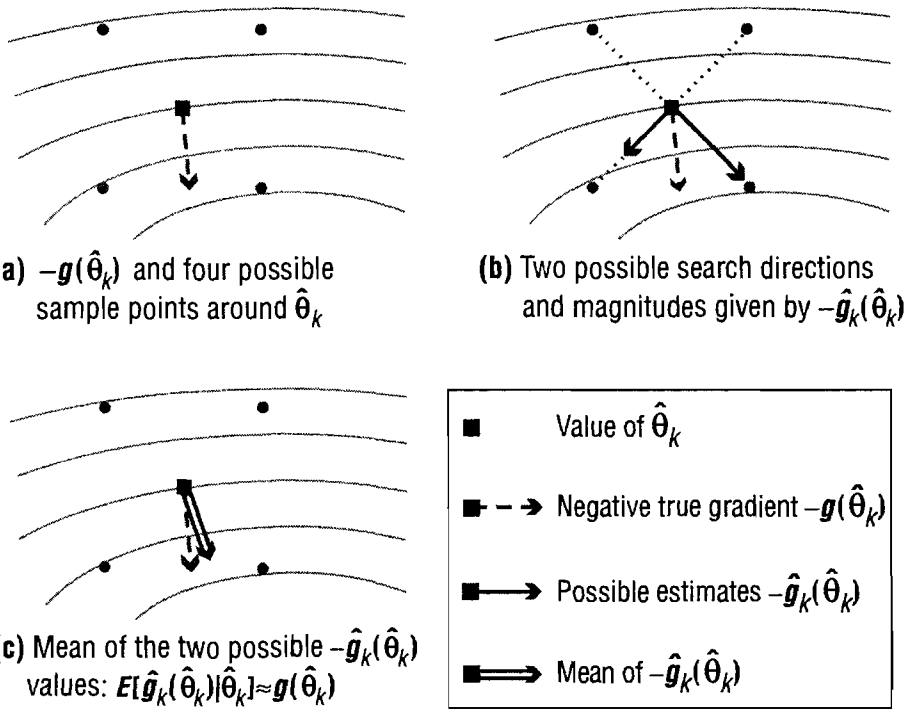

Figure 7.1. Comparisons of search direction and magnitude for the true gradient and SPSA gradient estimate in a low-noise setting with $p = 2$. Part (a) shows the four possible values of $\hat{\theta}_k \pm c_k\Delta_k$ surrounding $\hat{\theta}_k$ when using Bernoulli-distributed perturbations; true gradient is perpendicular to level curve at $\hat{\theta}_k$. Part (b) shows the two possible search directions and magnitudes for $-\hat{g}_k(\hat{\theta}_k)$. Each possibility has probability 1/2. The arrow pointing southeast is longer than the arrow pointing southwest because of the greater change in the function in sampling northwest-southeast points than in sampling southwest–northeast points. Part (c) shows the *mean* search direction and magnitude from the two possibilities in part (b). The slight deviation between the mean and true gradient is due to the bias.

Part (a) of Figure 7.1 shows the four points surrounding $\hat{\theta}_k$ (there are four *unique* possibilities for $\hat{\theta}_k \pm c_k\Delta_k$ from the four possible values for Δ_k). Note that the negative of the true gradient $-g(\hat{\theta}_k)$ points in a direction perpendicular to the level curve at $\hat{\theta}_k$ and in a direction of decreasing L (recall Section 1.4). Part (b) shows the negative of the two possible SP gradient estimates $-\hat{g}_k(\hat{\theta}_k)$ under the assumption that the noise (ε) is small. Note that one vector is longer than the other. This results from the difference in the loss values along the two lines forming the "X" in the figure. Along the northwest/southeast line, there is a greater change in the loss values than along the northeast/southwest line. From (7.2), it is apparent that the greater difference makes for a larger magnitude $-\hat{g}_k(\hat{\theta}_k)$. The difference in loss magnitudes may

be masked if the noise effects are relatively large (but it does not alter the *orientation* of the gradient estimates along the two lines in the "X" since the orientations depend solely on the evaluation points $\hat{\boldsymbol{\theta}}_k \pm c_k\boldsymbol{\Delta}_k$).

Finally, part (c) of Figure 7.1 shows the mean of the two possible negative gradient estimates for part (b) (which is one half of the vector sum of the two estimates in part (b)). This mean depends on *both* the orientation and the magnitude of the candidate vectors $-\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ in part (b). We see that the mean does not quite correspond to the true gradient. The indicated difference in $\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k)$ and $E[\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)\,|\,\hat{\boldsymbol{\theta}}_k]$ in part (c) represents the above-mentioned $O(c_k^2)$ bias.

In the manner of the FDSA bias discussion in Subsection 6.4.1, let us more formally analyze the bias $\boldsymbol{b}_k \equiv E\{[\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k) - \boldsymbol{g}(\hat{\boldsymbol{\theta}}_k)]\,|\,\Im_k\}$ where $\Im_k = \{\hat{\boldsymbol{\theta}}_0, \hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_k;\ \boldsymbol{\Delta}_0, \boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_{k-1}\}$ for $k \geq 1$. Relative to the definition of $\Im_k$ in Sections 4.3 and 6.4, this definition of $\Im_k$ is expanded to account for the additional randomness introduced through the $\boldsymbol{\Delta}_k$ process (which affects both the $\hat{\boldsymbol{\theta}}_k$ and noise ε processes). Let $L'''(\boldsymbol{\theta}) = \partial^3 L/\partial\boldsymbol{\theta}^T\partial\boldsymbol{\theta}^T\partial\boldsymbol{\theta}^T$ denote the $1 \times p^3$ row vector of all possible third derivatives of L. Assume that $L'''(\boldsymbol{\theta})$ exists and is continuous. By a Taylor expansion analogous to that in Section 6.4, the mth component of $\boldsymbol{b}_k$ is

$$b_{km} = \frac{1}{12}c_k^2 E\left\{\Delta_{km}^{-1}[L'''(\bar{\boldsymbol{\theta}}_k^{(+)}) + L'''(\bar{\boldsymbol{\theta}}_k^{(-)})][\boldsymbol{\Delta}_k \otimes \boldsymbol{\Delta}_k \otimes \boldsymbol{\Delta}_k]\,\middle|\,\Im_k\right\}, \qquad (7.5)$$

where $\bar{\boldsymbol{\theta}}_k^{(\pm)}$ denotes points on the (two) line segments between $\hat{\boldsymbol{\theta}}_k$ and $\hat{\boldsymbol{\theta}}_k \pm c_k\boldsymbol{\Delta}_k$ and $\otimes$ denotes the Kronecker product (Appendix A). Suppose that $|\Delta_{ki}| \leq \eta_0$, $E(|1/\Delta_{ki}|) \leq \eta_1$, and $|L'''_{i_1 i_2 i_3}(\boldsymbol{\theta})| \leq \eta_2$, where the η_i are positive constants and $L'''_{i_1 i_2 i_3}(\boldsymbol{\theta})$ denotes the element of $L'''(\boldsymbol{\theta})$ representing the third derivative with respect to the i_1, i_2, and i_3 elements of $\boldsymbol{\theta}$ ($i_j = 1, 2, \ldots, p$). Then, from the right-hand side of (7.5),

$$\begin{aligned} |b_{km}| &\leq \frac{\eta_2 c_k^2}{6}\sum_{i_1}\sum_{i_2}\sum_{i_3} E\left(\left|\frac{\Delta_{ki_1}\Delta_{ki_2}\Delta_{ki_3}}{\Delta_{km}}\right|\right) \\ &\leq \frac{\eta_2 c_k^2}{6}\left\{[p^3 - (p-1)^3]\eta_0^2 + (p-1)^3\eta_1\eta_0^3\right\} \end{aligned} \qquad (7.6)$$

(see Exercise 7.5). The bound in (7.6) provides an explicit form for the $O(c_k^2)$ bias.

7.3 BASIC ASSUMPTIONS AND SUPPORTING THEORY FOR CONVERGENCE

This section presents conditions for convergence of the SPSA iterate ($\hat{\boldsymbol{\theta}}_k \to \boldsymbol{\theta}^*$ a.s. as $k \to \infty$). The proof uses the ordinary differential equation (ODE) approach

discussed in Section 4.3 for the root-finding SA algorithm and Section 6.4 for the FDSA algorithm. The conditions here are close to the "engineering" conditions B.1′–B.5′ of Section 6.4 for FDSA. As with FDSA, but unlike stochastic gradient SA in Chapter 5, there are conditions on *two* gain sequences (a_k and c_k). In addition to the conditions for FDSA, however, we must impose conditions on the distribution of Δ_k, and the statistical relationship of Δ_k to the measurements $y(\cdot)$.

The conditions here ensuring convergence of $\hat{\theta}_k$ to a minimizing point θ^* are based on the arguments in Spall (1988b, 1992).[1] Following the pattern established in Section 6.4, condition B.*i*″ below is identical to, or closely related to, condition B.*i* in Section 4.3 and condition B.*i*′ in Section 6.4 for $i = 1, 2, \ldots, 5$ ($i = 6$ corresponds to a unique condition for SPSA). Let $\varepsilon_k^{(\pm)} = \varepsilon(\hat{\theta}_k \pm c_k\Delta_k)$. When a condition is identical to one of the previous conditions in Sections 4.3 and/or 6.4 and relatively long to state, we simply refer back to the earlier condition rather than restate the condition here.

B.1″ **(Gain sequences)** Same as condition B.1′ in Section 6.4 (i.e., a_k and $c_k > 0$; a_k and $c_k \to 0$; $\sum_{k=0}^{\infty} a_k = \infty$; and $\sum_{k=0}^{\infty} a_k^2/c_k^2 < \infty$).

B.2″ **(Relationship to ODE)** Same as condition B.2 in Section 4.3 (and condition B.2′ in Section 6.4).

B.3″ **(Iterate boundedness)** Same as condition B.3 in Section 4.3 (and condition B.3′ in Section 6.4). Main requirement: $\sup_{k\geq 0}\|\hat{\theta}_k\| < \infty$ a.s.

B.4″ **(Measurement noise; relationship between the measurement noise and Δ_k)** For all k, $E[(\varepsilon_k^{(+)} - \varepsilon_k^{(-)})|\mathfrak{I}_k, \Delta_k] = 0$ and the ratio of measurement to perturbation is such that $E\left[\left(y(\hat{\theta}_k \pm c_k\Delta_k)/\Delta_{ki}\right)^2\right]$ is uniformly bounded (over k and i).

B.5″ **(Smoothness of L)** L is three-times continuously differentiable and bounded on $\mathbb{R}^p$. (This is slightly stronger than condition B.5′ in Section 6.4, which pertains only to the "unmixed" partials $L'''_{iii}(\theta)$ for all i.)

B.6″ **(Statistical properties of the perturbations)** The $\{\Delta_{ki}\}$ are independent for all k, i, identically distributed for all i at each k, symmetrically distributed about zero and uniformly bounded in magnitude for all k, i.

Let us comment on the above conditions. From the point of view of the user's input, conditions B.1″, B.4″, and B.6″ are the most relevant since they govern the gains a_k, c_k and the random perturbations Δ_k. The role of a_k in B.1″ is similar to its role in the root-finding SA algorithm, as discussed in Section 4.3. The square summability in condition B.1″ ($\sum_{k=0}^{\infty} a_k^2/c_k^2 < \infty$) balances the decay

[1] The conditions given here differ slightly from those in Spall (1988b, 1992). The conditions here are streamlined to be closer to the minimal required using the proof techniques in Spall (1988b, 1992).

of a_k against the decay of c_k to ensure that the update in moving $\hat{\boldsymbol{\theta}}_k$ to $\hat{\boldsymbol{\theta}}_{k+1}$ is well behaved. In particular, the condition prevents c_k from going to zero too quickly, which prevents the gradient estimate from becoming too wild and overpowering the decay associated with a_k. The motivation behind B.2″ and B.3″ is identical to the motivation for B.2 and B.3 in Section 4.3. These conditions impose the requirement that $\hat{\boldsymbol{\theta}}_k$ (including the initial condition) is close enough to $\boldsymbol{\theta}^*$ so that there is a natural tendency for an analogous deterministic algorithm (manifested in continuous time as an ODE) to converge to $\boldsymbol{\theta}^*$.

Conditions B.4″ to B.6″ on the perturbation distribution and smoothness of L guarantee that the gradient estimate $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ is an unbiased estimate of $\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k)$ to within an $O(c_k^2)$ error. This $O(c_k^2)$ bias is small enough so that (as in the stochastic gradient case, where the gradient estimate typically has no bias) the $\boldsymbol{\theta}$ iterate is able to converge to $\boldsymbol{\theta}^*$. Further, the boundedness of $E\left[\left(y(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k)/\Delta_{ki}\right)^2\right]$ in B.4″ is used together with $\sum_{k=0}^{\infty} a_k^2/c_k^2 < \infty$ in B.1″ to ensure that a sum of variances is finite, analogous to the proof of convergence of FDSA.

Let us comment on the important relationship of finite inverse moments for the elements of Δ_k to the condition in B.4″ that $E\left[\left(y(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k)/\Delta_{ki}\right)^2\right]$ be bounded. Using Hölder's inequality (see Exercise C.4, Appendix C),

$$E\left[\left(\frac{y(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k)}{\Delta_{ki}}\right)^2\right] \le \left[E\left(\left|y(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k)\right|^{2+2\eta}\right)\right]^{1/(1+\eta)}\left[E\left(\left|\frac{1}{\Delta_{ki}}\right|^{2+2\tau}\right)\right]^{1/(1+\tau)}, \tag{7.7}$$

where η and τ are any strictly positive values that satisfy $(1+\eta)^{-1}+(1+\tau)^{-1} = 1$ (see Exercise 7.6). The analyst may choose η and τ arbitrarily subject to these conditions. If the measurements $y(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k)$ have bounded moments of order $2+2\eta$, then the first $[\cdot]$ term on the right-hand side of (7.7) is bounded. Hence, (7.7) implies that $E\left[\left(y(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k)/\Delta_{ki}\right)^2\right]$ is uniformly bounded if there exists a $\tau > 0$ such that $(1+\eta)^{-1}+(1+\tau)^{-1} = 1$ and

$$E\left(\left|\frac{1}{\Delta_{ki}}\right|^{2+2\tau}\right) \le C \tag{7.8}$$

for some $C > 0$. This bounded inverse moments condition for the Δ_{ki} is an important part of SPSA. We saw in the informal arguments of (7.3) and (7.4) an application of a similar (weaker) condition in showing that $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ is a nearly unbiased estimate of $\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k)$.

One important—and very simple—distribution that satisfies the inverse moments condition is the symmetric Bernoulli ± 1 distribution. Two common

mean-zero distributions that do *not* satisfy the inverse moments condition are symmetric uniform and normal with mean zero. The failure of both of these distributions is a consequence of the amount of probability mass near zero. In particular, with too much probability near zero, the expectation integral of the generic form

$$E\left(\left|\frac{1}{\Delta_{ki}}\right|^{2+2\tau}\right) = \int_{-\infty}^{\infty} \left|\frac{1}{\delta}\right|^{2+2\tau} p_{\Delta}(\delta)\, d\delta \tag{7.9}$$

is undefined (infinite), where $p_{\Delta}(\delta)$ represents the density function for Δ_{ki} (or mass function if the integral is replaced by a sum).

Figure 7.2 shows several probability density or probability mass functions for mean-zero random variables. The top row shows density or mass functions for which the inverse moments $E\left(\left|1/\Delta_{ki}\right|^{2+2\tau}\right)$ are finite; the bottom row shows functions for which the inverse moments are not defined. Although the U-shaped density function is shown in the top row, an arbitrary U-shaped density may not have a finite inverse $2 + 2\tau$ moment, depending on the curvature of the "U" and the value of τ. One must evaluate (7.9) to determine if it is finite (Exercise 7.8).

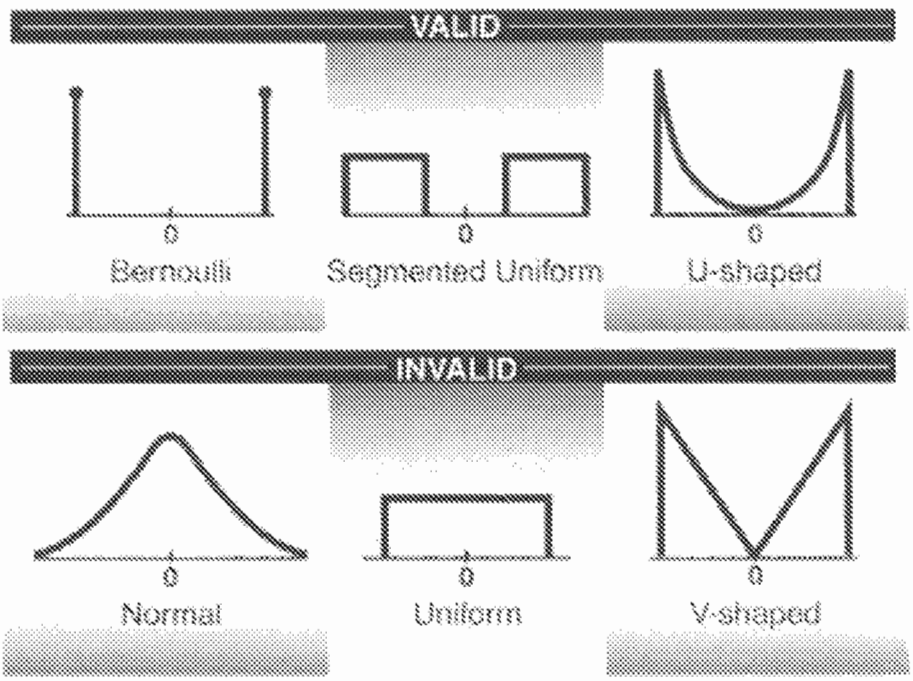

Figure 7.2. Probability density or mass functions that are valid or invalid relative to the inverse moments condition: $E\left(\left|1/\Delta_{ki}\right|^{2+2\tau}\right) < \infty$ for any k, i.

Aside from Spall (1992), conditions for convergence of SPSA have been presented in Dippon and Renz (1997), Wang and Chong (1998), Chen et al. (1999), and Gerencsér (1999). While the conditions of these authors tend to be slightly different from one another, in all cases there are conditions similar to the finite inverse moments conditions discussed above for the elements of Δ_k. The theorem below is proved in Spall (1992).

Theorem 7.1. Suppose that conditions B.1″–B.6″ hold. Further, suppose that $\boldsymbol{\theta}^*$ is a unique minimum (i.e., Θ^* is the singleton $\boldsymbol{\theta}^*$). Then, for the SPSA algorithm in (7.1), $\hat{\boldsymbol{\theta}}_k \to \boldsymbol{\theta}^*$ a.s. as $k \to \infty$.

7.4 ASYMPTOTIC NORMALITY AND EFFICIENCY ANALYSIS

Although the convergence result for SPSA is of some independent interest, the most interesting theoretical results in Spall (1992), and those that provide most of the rationale for using SPSA, are the asymptotic efficiency conclusions that follow from an asymptotic normality result. In particular, in addition to conditions B.1″–B.6″ above, let us suppose that B.7″–B.9″ in the chapter appendix (Section 7.10) hold. The theorem below is proved in Spall (1992).

Theorem 7.2. Suppose that the gains have the standard form $a_k = a/(k+1+A)^\alpha$ and $c_k = c/(k+1)^\gamma$, $k = 0, 1, 2,\ldots$, with a, c, α, and γ strictly positive, $A \geq 0$, $\beta = \alpha - 2\gamma > 0$ (as in Section 6.5), and $3\gamma - \alpha/2 \geq 0$. Further, suppose that the conditions of Theorem 7.1 plus conditions B.7″ – B.9″ hold. Then, for the SPSA algorithm in (7.1),

$$k^{\beta/2}(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*) \xrightarrow{\text{dist.}} N(\boldsymbol{\mu}_{SP}, \boldsymbol{\Sigma}_{SP}) \text{ as } k \to \infty, \tag{7.10}$$

where $\boldsymbol{\mu}_{SP}$ and $\boldsymbol{\Sigma}_{SP}$ are a mean vector and covariance matrix.

In Theorem 7.2, $\boldsymbol{\mu}_{SP}$ depends on both the Hessian matrix and the third derivatives of $L(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^*$ and $\boldsymbol{\Sigma}_{SP}$ depends on the Hessian matrix at $\boldsymbol{\theta}^*$ (notation consistent with $\boldsymbol{\mu}_{FD}$ and $\boldsymbol{\Sigma}_{FD}$ from Section 6.5). The specific forms are given in Spall (1992) and Dippon and Renz (1997). In general, $\boldsymbol{\mu}_{SP} \neq \boldsymbol{\mu}_{FD}$ and $\boldsymbol{\Sigma}_{SP} \neq \boldsymbol{\Sigma}_{FD}$. As in the FDSA result in Section 6.5, $\boldsymbol{\mu}_{SP} \neq \mathbf{0}$ in general, which contrasts with many well-known asymptotic normality results in estimation, including that in Section 4.4 for the root-finding SA algorithm. As discussed in Section 6.5 for FDSA, $\boldsymbol{\mu}_{SP} \neq \mathbf{0}$ does *not* imply that $\hat{\boldsymbol{\theta}}_k$ is an asymptotically biased estimator.

Similar to the root-finding SA and FDSA cases, the asymptotic distribution result (7.10) allows one to determine asymptotically optimal gain decay rates (i.e., rates that provide the maximum value of $\beta/2$). Given the restrictions in the theorem statement (including B.1″), these are $\alpha = 1$ and $\gamma = 1/6$, yielding $\beta/2 = 1/3$ (the same as FDSA—see Section 6.5). Hence, the fastest

possible stochastic rate at which the error $\hat{\theta}_k - \theta^*$ goes to zero is proportional to $k^{-\beta/2} = 1/k^{1/3}$ for large k. This contrasts with the fastest allowable rate (proportional to $1/\sqrt{k}$) for the root-finding (stochastic gradient) SA algorithm.

Hence, one measure of the value of the gradient information in stochastic gradient SA is the increase in rate of convergence. (Section 14.4 discusses a special case where it also possible to get a $1/\sqrt{k}$ rate of convergence in SPSA through the use of common random numbers in a simulation-based optimization context.) However, as discussed in Section 6.6, it is generally superior in finite-sample practice to have gain sequences a_k and c_k that decay more slowly than the asymptotically optimal gains using $\alpha = 1$ and $\gamma = 1/6$. As with FDSA, $\alpha = 0.602$ and $\gamma = 0.101$ are approximately the lowest possible valid values and are recommended in many practical applications (see the implementation guidelines in Section 7.5).

Spall (1992, Sect. 4) uses the asymptotic normality result in (7.10) (together with the parallel result for FDSA in Section 6.5) to evaluate the relative efficiency of SPSA. This efficiency depends on the shape of L, the values for a_k and c_k, and the distributions of the Δ_{ki} and measurement noise terms $\varepsilon_k^{(\pm)}$. There is no single expression that can be used to characterize the relative efficiency.

As discussed in Spall (1992, Sect. 4) and Chin (1997), however, in most practical problems, SPSA will be asymptotically more efficient than FDSA. For example, if $3\gamma - \alpha/2 > 0$ (as in the guidelines in Section 7.5, with $\alpha = 0.602$ and $\gamma = 0.101$), then by equating the asymptotic mean-squared errors of the parameter estimates, as given by the asymptotic distributions in FDSA and SPSA, we find that

$$\boxed{\frac{\text{number of } y(\theta) \text{ values in SPSA}}{\text{number of } y(\theta) \text{ values in FDSA}} \to \frac{1}{p}} \tag{7.11}$$

as the number of loss measurements in both procedures gets large. (The above result also sometimes applies with the asymptotically optimal condition $3\gamma - \alpha/2 = 0$ holding; see Spall, 1992.) Expression (7.11) implies that the p-fold savings per iteration (per gradient approximation) translates directly into a p-fold savings in the overall optimization process. Note that (7.11) is derived under the assumption that FDSA and SPSA use the same gain sequences a_k and c_k (including, e.g., sequences tuned for optimal FDSA performance).

The above efficiency result is derived under the assumption of noisy loss measurements. Perhaps counterintuitively, the mathematics for efficiency analysis in the noise-free case is more difficult than for the noisy case (it is *not* valid to simply take the limit of the relevant expressions in the noisy case as the noise variance goes to zero). Gerencsér and Vágó (2001) analyze the efficiency in the noise-free case and find a result analogous to the convergence rate for deterministic steepest descent algorithm, as in Section 1.4 (i.e., $\|\hat{\theta}_{k+1} - \theta^*\| =$

$O(\|\hat{\theta}_k - \theta^*\|)$ a.s., or, equivalently, $\|\hat{\theta}_{k+1} - \theta^*\| = O(\lambda^k)$ a.s. for some $0 < \lambda < 1$). This rate is superior to the above-mentioned $1/k^{1/3}$ rate in the noisy case.

By providing theoretical evidence of the superiority of one algorithm over another, it might seem that (7.11) contradicts the no free lunch (NFL) theorems of Subsection 1.2.2. It does not. Recall that the NFL theorems state that the performance of any two algorithms is the same when averaged across *all possible problems*. The asymptotic superiority of SPSA over FDSA as expressed in (7.11) applies to only a sliver of all possible problems—those problems satisfying the conditions of the asymptotic normality. Of course, this sliver is an important practical subset of all possible problems, but, nonetheless, it is only a subset.

7.5 PRACTICAL IMPLEMENTATION

7.5.1 Step-by-Step Implementation

The step-by-step summary below shows how SPSA iteratively produces a sequence of estimates. MATLAB code for implementing the steps below is available at the book's Web site.

Basic SPSA Algorithm

Step 0 **(Initialization and coefficient selection)** Set counter index $k = 0$. Pick initial guess $\hat{\theta}_0$ and nonnegative coefficients a, c, A, α, and γ in the SPSA gain sequences $a_k = a/(k+1+A)^\alpha$ and $c_k = c/(k+1)^\gamma$. Practically effective (and theoretically valid) values for α and γ are 0.602 and 0.101, respectively; a, A, and c may be determined based on the practical guidelines given in Subsection 7.5.2.

Step 1 **(Generation of the simultaneous perturbation vector)** Generate by Monte Carlo a p-dimensional random perturbation vector Δ_k, where each of the p components of Δ_k are independently generated from a zero-mean probability distribution satisfying the conditions above. An effective (and theoretically valid) choice for each component of Δ_k is to use a Bernoulli ± 1 distribution with probability of 1/2 for each ± 1 outcome, although other choices are valid and may be desirable in some applications.

Step 2 **(Loss function evaluations)** Obtain two measurements of the loss function based on the simultaneous perturbation around the current $\hat{\theta}_k$: $y(\hat{\theta}_k + c_k\Delta_k)$ and $y(\hat{\theta}_k - c_k\Delta_k)$ with the c_k and Δ_k from steps 0 and 1.

Step 3 **(Gradient approximation)** Generate the simultaneous perturbation approximation to the unknown gradient $g(\hat{\theta}_k)$ according to eqn. (7.2). It is sometimes useful to average several gradient approximations at $\hat{\theta}_k$, each formed from an independent generation of Δ_k. The benefits are especially apparent if the noise effects ε_k are relatively large.

Step 4 **(Update θ estimate)** Use the standard SA form in (7.1) to update $\hat{\theta}_k$ to a new value $\hat{\theta}_{k+1}$. Check for constraint violation (if relevant) and modify the updated θ. (A common way to handle "easy" constraints is to simply map violating elements of θ to the nearest valid point.)

Step 5 **(Iteration or termination)** Return to step 1 with $k+1$ replacing k. Terminate the algorithm if there is little change in several successive iterates or if the maximum allowable number of iterations has been reached.

In addition to the practical guidelines above, the blocking steps discussed in Subsection 7.8.2 for the adaptive SPSA approach can also be applied. In these steps, the iteration update is blocked (i.e., $\hat{\theta}_{k+1}$ is set to $\hat{\theta}_k$) if there would otherwise be a suspiciously large change in θ or if the measured loss value at the intended value $\hat{\theta}_{k+1}$ does not show enough improvement relative to the value at $\hat{\theta}_k$. Blocking based on an excessive change in θ requires no additional loss measurements; blocking based on a check of the loss value requires at least one additional loss measurement per iteration (i.e., measurement(s) at the nonperturbed values $\hat{\theta}_k$ versus measurements at only the perturbed values $\hat{\theta}_k \pm c_k\Delta_k$).

A further practical concern is to attempt to define θ so that the magnitudes of the θ elements are similar to one another. This desire is apparent by noting that the magnitudes of all components in the perturbations $c_k\Delta_k$ are identical in the case where Bernoulli perturbations are used. By defining the elements in θ such that they are of similar magnitude, it is possible to ensure that $c_k\Delta_k$ is being added and subtracted to a vector where all components have a similar magnitude in the course of the search process. Although not always possible, an analyst often has the flexibility to choose the units for θ to ensure similar magnitudes. Consider the following example.

Example 7.1—Definition of θ for an electrical circuit. Consider the optimization of an electrical circuit with variable resistors, inductors, and capacitors. The elements of θ are the resistance, inductance, and capacitance of the components in the circuit. Suppose that the "typical" magnitudes of the resistance, inductance, and capacitance of the components are 5 to 20 ohms, 2 to 10 millihenries, and 3×10^{-6} to 15×10^{-6} farads. It is undesirable to have a θ vector with some components having magnitudes of order 10^0 to 10^1 and other components having magnitude of order 10^{-6}. If we define the units of θ to be ohms, millihenries, and microfarads, then the magnitude of all components of θ will be commensurate at between 2 and 20. ❑

7.5.2 Choice of Gain Sequences

Let us summarize some additional implementation aspects regarding the choice of algorithm gain sequences a_k, c_k. The reader should be warned that the

guidelines provided here are just that—guidelines—and may not be the best for every application. These guidelines were developed based on many test cases conducted by the author and others and form a reasonable basis for starting if one has no specific reason to follow other guidelines. (Theoretical guidelines, such as discussed in Fabian, 1971, and Chin, 1997, are not generally useful in practical applications since they require the very information on the loss function and its gradients that is assumed unavailable!) The guidelines here are a natural extension of those in Section 6.6 for FDSA.

The choice of the gain sequences is critical to the performance. With α and γ as specified in step 0 of the algorithm above, one typically finds that in a high-noise setting (i.e., poor quality measurements of L) it is necessary to pick a smaller a and larger c than in a low-noise setting. As noted below Theorem 7.2, the asymptotically optimal values of α and γ with noisy loss measurements are 1 and 1/6, respectively. In practice, however, it is usually the case that $\alpha < 1$ yields better finite-sample performance through maintaining a larger step size. Hence the recommendation in step 0 to use values (0.602 and 0.101) that are effectively the lowest allowable subject to satisfying the theoretical conditions mentioned in Sections 7.3 and 7.4. When the algorithm is being run with a large number of iterations, it may be beneficial to convert to $\alpha = 1$ and $\gamma = 1/6$ at some point in the iteration process to take advantage of the asymptotic optimality.

With the Bernoulli ± 1 distribution for the elements of Δ_k and the α and γ specified in step 0, a rule of thumb is to set c at a level approximately equal to the standard deviation of the measurement noise in $y(\boldsymbol{\theta})$. This helps keep the p elements of $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ from getting excessively large in magnitude. The standard deviation can be estimated by collecting several $y(\boldsymbol{\theta})$ values at the initial guess $\hat{\boldsymbol{\theta}}_0$. When perfect (noise-free) measurements of $L(\boldsymbol{\theta})$ are available, then c should be chosen as some small positive number.

The values of a, A can be chosen together to ensure effective practical performance of the algorithm. A useful rule of thumb is to choose $A > 0$ such that it is 10 percent or less of the maximum number of expected/allowed iterations. After choosing A, one can choose a such that $a_0 = a/(1+A)^{0.602}$ times the magnitude of elements in $\hat{\boldsymbol{g}}_0(\hat{\boldsymbol{\theta}}_0)$ is approximately equal to the smallest of the desired change magnitudes among the elements of $\boldsymbol{\theta}$ in the early iterations. To do this reliably may require several replications of $\hat{\boldsymbol{g}}_0(\hat{\boldsymbol{\theta}}_0)$. These guidelines for choosing a are similar to those mentioned in Brennan and Rogers (1995, Sect. 2). An example of gain selection is given below.

Example 7.2—Choice of gain coefficients. Suppose that the standard deviation of the noise $\varepsilon_k(\boldsymbol{\theta})$ at $\boldsymbol{\theta}$ near $\hat{\boldsymbol{\theta}}_0$ is approximately 0.5 and that there is a budget of 2000 loss measurements for the search process. This suggests a choice of $c = 0.5$ and $A = 0.10 \times 2000/2 = 100$ (10 percent of $2000/2 = 1000$ iterations). Based on prior information, suppose that the analyst felt that the elements of $\boldsymbol{\theta}$ should typically move by a magnitude 0.1 in the early iterations. After computing

several values of $\hat{g}_0(\hat{\theta}_0)$, it is determined that the mean magnitude of the elements in $\hat{g}_0(\hat{\theta}_0)$ (after choosing c as above) is approximately 10. With $A = 100$, it is found that $a = 0.16$ according to $[0.16/(100+1)^{0.602}]\times 10 = 0.1$. ❑

7.6 NUMERICAL EXAMPLES

This section presents two examples illustrating the relative efficiency of FDSA and SPSA, including the implications of the theoretical comparison in Section 7.4 (especially (7.11)).

Example 7.3—FDSA versus SPSA on Rosenbrock function. Consider a $p = 10$ version of the Rosenbrock test function introduced in Section 2.2. The function has the fourth-order polynomial form

$$L(\theta) = \sum_{i=1}^{5}\left[100(t_{2i} - t_{2i-1}^2)^2 + (1 - t_{2i-1})^2\right],$$

where $\theta = [t_1, t_2, \ldots, t_{10}]^T$. Note that $L(\theta^*) = 0$ at $\theta^* = [1, 1, \ldots, 1]^T$. Assume that the function measurements are taken in the presence of independent, identically distributed (i.i.d.) noise having distribution $N(0, 0.2^2)$.

Given the desire to test the numerical implications of the *asymptotic* theory in Section 7.4, we consider an initial condition that is close to θ^* (appropriate since it is assumed that the algorithms have been running a long time). Let $\hat{\theta}_0 = [0.99, 1, 0.99, 1, \ldots, 1]^T$, so $L(\hat{\theta}_0) = 0.1985$. Note that the "one-sigma" value of the noise is slightly greater than the initial loss value and will dominate the $L(\theta)$ information contained in $y(\theta)$ if (as one hopes!) the iterate moves closer to θ^*. Letting $A = 10$ and $c = 0.05$ in the standard gains $a_k = a/(k+1+A)^{0.602}$ and $c_k = c/(k+1)^{0.101}$, we find numerically that $a = 0.002$ is an approximately optimal value for FDSA when using 10,000 to 50,000 loss measurements (the A and c values are smaller than the values suggested in Subsection 7.5.2 to better emulate asymptotic effects). Consistent with the efficiency result in Section 7.4, both FDSA and SPSA are run with the same gain values.

Table 7.1 shows the mean values of the normalized loss function

$$L_{\text{norm}}(\theta) \equiv \frac{L(\theta) - L(\theta^*)}{L(\hat{\theta}_0) - L(\theta^*)}$$

over 50 independent pairs of FDSA and SPSA runs for several run lengths. Note that with the relatively high noise level and closeness of $\hat{\theta}_0$ and θ^*, there is a significant challenge to the algorithms to produce notably improved values of θ

Table 7.1. Sample means of normalized loss $L_{\text{norm}} = L_{\text{norm}}(\hat{\theta}_k)$ at terminal $\hat{\theta}_k$ for FDSA and SPSA over 50 independent replications. Number of loss measurements $y(\theta)$ is such that FDSA and SPSA take the same number of iterations in each comparison.

Number of $y(\theta)$ values [*number of iterations*]	Mean L_{norm} for FDSA	Mean L_{norm} for SPSA
1000-FDSA; 100-SPSA [*50 iterations*]	0.100	0.111
25,000-FDSA; 2500-SPSA [*1250 iterations*]	0.0014	0.0017
50,000-FDSA; 5000-SPSA [*2500 iterations*]	0.0012	0.0011

that manifest themselves in low values of $L_{\text{norm}}(\theta)$. The value for θ used in the 50 evaluations of $L_{\text{norm}}(\theta)$ for each of FDSA and SPSA is the value at the final iteration after the indicated number of loss measurements. Table 7.1 shows three combinations of number of loss evaluations. The ratio of number of loss evaluations in FDSA to SPSA is equal to p, so that the algorithms use the same number of iterations (50, 1250, or 2500).

In the three cases of Table 7.1, the sample means for the terminal $L_{\text{norm}}(\hat{\theta}_k)$ values are relatively close for FDSA and SPSA even though the common gain sequences are tuned to approximately optimize performance for FDSA. This provides numerical support for the efficiency result in Section 7.4. That is, even with the ten-fold savings in number of loss measurements in SPSA per iteration, FDSA and SPSA perform comparably with the same number of iterations. ❑

Example 7.4—Neural network control. Figure 7.3 shows results for a $p = 412$ problem in the control of a wastewater treatment system (see Spall and Cristion, 1997 and 1998, for a complete description of the problem and the results of similar studies). The overall approach is the model-free controller summarized in Section 6.2. A feedforward neural network (Section 5.2) with $p = 412$ weights is used for the control function. FDSA and SPSA are used to estimate the weights. The curves represent an average of 50 independent realizations (an individual realization, of course, is much more jagged than the smooth curves shown in the figure.) The loss function L_k is time-varying, reflecting varying target values for water cleanliness and methane gas. The gain coefficients in FDSA and SPSA are separately tuned to approximately optimize the performance of each algorithm.

Note that FDSA and SPSA perform comparably on an iteration-by-iteration basis. The compelling part of the story, however, is that SPSA uses only two measurements per iteration while FDSA uses 824. This 412-fold savings in noisy loss measurements per iteration leads to the large savings in total

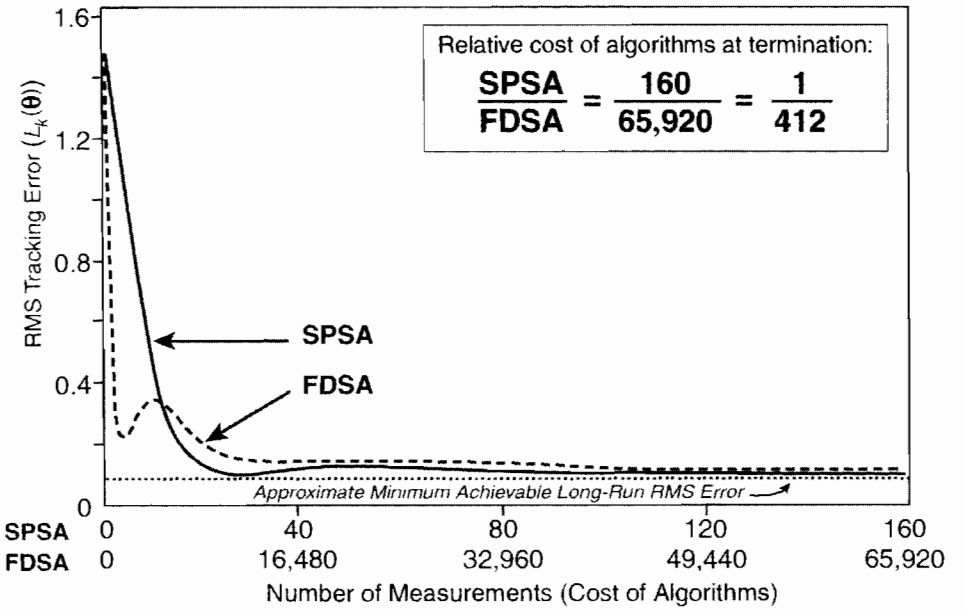

Figure 7.3. Comparison of efficiency for FDSA and SPSA in wastewater treatment problem. Points along the horizontal axis correspond to a common number of iterations for both FDSA and SPSA; 80 iterations are shown. Curves represent sample mean of 50 independent replications.

measurements for the full number of iterations, as shown in the box in the upper right corner. Note that this example is not a direct illustration of efficiency result (7.11) in Section 7.4 because of the time-varying loss functions and noncommon gains. Nevertheless, the relative performance here is essentially the same as predicted in (7.11). ❑

7.7 SOME EXTENSIONS: OPTIMAL PERTURBATION DISTRIBUTION; ONE-MEASUREMENT FORM; GLOBAL, DISCRETE, AND CONSTRAINED OPTIMIZATION

Sadegh and Spall (1998) consider the problem of choosing the best distribution for the Δ_k vector. Based on the asymptotic distribution result in Section 7.4, it is shown that the optimal distribution for the components of Δ_k is symmetric Bernoulli. This simple distribution has also proven effective in many finite-sample practical and simulation examples. The recommendation in step 1 of the algorithm description in Section 7.5 follows from these findings. It should be noted, however, that other distributions are sometimes desirable. Because the user has full control over this choice and since the generation of Δ_k represents a trivial cost toward the optimization, it may be worth evaluating other

possibilities in some applications. For example, Maeda and De Figueiredo (1997) use a symmetric segmented uniform distribution (i.e., a uniform distribution with a section removed near zero to preserve the finiteness of inverse moments), in an application for robot control (this density function is depicted in Figure 7.2).

A *one*-measurement form of the SP gradient approximation is considered in Spall (1997). The gradient approximation has the form

$$\hat{g}_k(\hat{\theta}_k) = \begin{bmatrix} \dfrac{y(\hat{\theta}_k + c_k\Delta_k)}{c_k\Delta_{k1}} \\ \vdots \\ \dfrac{y(\hat{\theta}_k + c_k\Delta_k)}{c_k\Delta_{kp}} \end{bmatrix}. \tag{7.12}$$

Although the form above may seem strange in that it does not include explicit information related to the difference of function values, the form shares the nearly unbiased property of the standard two-measurement form in (7.2). In particular, via a Taylor expansion of $L(\hat{\theta}_k + c_k\Delta_k)$, it is found that $E[\hat{g}_k(\hat{\theta}_k)|\hat{\theta}_k] = g(\hat{\theta}_k) + O(c_k^2)$ (Exercise 7.12). Although it is shown in Spall (1997) that the standard two-measurement form is usually more efficient (in terms of the total number of loss function measurements to obtain a given level of accuracy in the θ iterate), there may be advantages to the one-measurement form in real-time operations. Such real-time applications include target tracking and feedback control, where the underlying system dynamics may change too rapidly to get a credible gradient estimate with two successive measurements.

There are several types of averaging that have been used in the context of SPSA. Dippon and Renz (1997) explore *iterate* averaging (analogous to the idea discussed in Subsection 4.5.3 for root-finding SA), showing that the approach can achieve near-optimal asymptotic mean-squared errors for the iterate average $\bar{\theta}_k$. However, as discussed in Spall (2000), this approach may perform relatively poorly in practical finite-sample problems. This is also discussed in Subsection 4.5.3 in the context of root-finding SA.

Aside from iterate averaging, two methods for averaging the gradient estimate $\hat{g}_k(\hat{\theta}_k)$ have been considered with the aim of reducing the variability of the input at each iteration. The first is simply to average several gradient approximations at *each iteration* (at the cost of additional function measurements). This is mentioned in step 3 of Subsection 7.5.1 and discussed in Spall (1992). The other method (discussed in Spall and Cristion, 1994) is gradient smoothing in a manner analogous to momentum in the neural network literature (momentum is discussed in Section 5.2 in the context of the backpropagation algorithm). Both the simple averaging and the smoothing ideas

are aimed at coping with inaccuracies in the gradient estimate resulting from the simultaneous perturbation aspect and the noise in the loss measurements.

The use of SPSA for *global* minimization among multiple local minima is discussed in Maryak and Chin (2001). One of their approaches relies on injecting Monte Carlo noise in the right-hand side of the basic SPSA updating step in (7.1). This approach is a common way of converting SA algorithms to global optimizers (Yin, 1999). Maryak and Chin (2001) also show that basic SPSA *without* injected noise (i.e., eqn. (7.1) without modification) may, under certain conditions, be a global optimizer. Formal justification for this important result follows because the random error in the SP gradient approximation acts in a way that is statistically equivalent to the injected noise mentioned above. Although the injected noise approach is relatively well known, basic SPSA as a global optimizer has a faster asymptotic rate of convergence and reduces the number of user-specified coefficients. Section 8.4 considers in greater detail the subject of global optimization via SA, including the use of SPSA without injected noise.

Discrete optimization problems (where $\boldsymbol{\theta}$ may take on discrete or combined discrete/continuous values) are discussed in Gerencsér et al. (1999). Discrete SPSA relies on a fixed-gain (constant a_k and c_k) version of the standard SPSA method. The loss function is assumed to be convex and, in the process of optimization, is temporarily extended to a unique, continuous convex function. The continuous extension is then used to form a gradient approximation, which is used in a fixed-gain SA algorithm. The parameter estimates produced are constrained to lie on the discrete-valued grid.

The problem of constrained (equality and inequality) optimization with SPSA is considered in Sadegh (1997) and Fu and Hill (1997) using a projection approach. The projection algorithm is a direct analogue of the constrained root-finding SA algorithm in Section 4.1:

$$\hat{\boldsymbol{\theta}}_{k+1} = \Psi_{\Theta}[\hat{\boldsymbol{\theta}}_k - a_k \hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)],$$

where $\Psi_{\Theta}[\cdot]$ is the mapping that projects any point not in the constraint domain Θ to a new point inside Θ. While the projection approach has an elegant mathematical form, it is quite restricted in the types of constraints that can be handled in practical problems. Essentially, the constraints must be represented explicitly in a "nice" way so as to facilitate the mapping of a constraint violation in $\boldsymbol{\theta}$ to the nearest valid point. A common implementation of projections is to problems with hypercube constraints, where the individual components of $\boldsymbol{\theta}$ are bounded above and below by user-specified constants.

An alternative approach to constrained optimization is given in Wang and Spall (1999). This approach is based on altering the loss function to include a penalty term. In particular, at iteration k, $L(\boldsymbol{\theta})$ is replaced by a modified loss function

$$L(\boldsymbol{\theta}) + r_k P(\boldsymbol{\theta}),$$

where r_k is an increasing sequence of positive scalars ($r_k \to \infty$) and $P(\boldsymbol{\theta})$ is a penalty function that takes on large positive values when the constraints are violated. In many practical problems constraints are only implicit in $\boldsymbol{\theta}$, and the penalty function approach is well suited to handle such cases. For example, if it is required that $0 \leq f(\boldsymbol{\theta}) \leq 1$ for some function $f(\cdot)$, then $P(\boldsymbol{\theta})$ can be designed to take on very large values when $\boldsymbol{\theta}$ is such that $f(\boldsymbol{\theta})$ is outside of [0, 1]. This does not require explicit constraints on the components of $\boldsymbol{\theta}$. Although the penalty function method has broad applicability, the implementation is sometimes a challenge. In general, the specific choice of r_k and $P(\boldsymbol{\theta})$ dramatically affect the performance of the method.

7.8 ADAPTIVE SPSA

We now discuss an adaptive approach based on estimating the Hessian matrix of $L(\boldsymbol{\theta})$ (or, equivalently, the Jacobian matrix of $\boldsymbol{g}(\boldsymbol{\theta})$). The three subsections in this section introduce the adaptive algorithm, discuss some practical implementation issues, and summarize the theory on efficiency.

7.8.1 Introduction and Basic Algorithm

Using the simultaneous perturbation idea, this section presents a general adaptive SPSA (ASP) approach that is based on a simple method for estimating the Hessian (or Jacobian) matrix while, concurrently, estimating the primary parameters of interest ($\boldsymbol{\theta}$). The ASP approach produces a stochastic analogue to the deterministic Newton–Raphson algorithm considered in Section 1.4, leading to a recursion that is optimal or near-optimal in its rate of convergence and asymptotic error. The approach applies in both the gradient-free setting emphasized in this chapter and in the root-finding/stochastic gradient-based (Robbins–Monro) setting considered in Chapters 4 and 5. Like the standard (first-order) SPSA algorithm, the ASP algorithm requires only a small number of loss function (or gradient, if relevant) measurements per iteration—independent of the problem dimension—to adaptively estimate the Hessian and parameters of primary interest.

There are other second-order SA approaches. A more complete discussion on related work is given in Spall (2000). In the gradient-free setting, Fabian (1971) forms estimates of the gradient and Hessian for a Newton–Raphson-type SA algorithm by using, respectively, a finite-difference approximation and a set of differences of finite-difference approximations. This leads to $O(p^2)$ measurements of L per update of the $\boldsymbol{\theta}$ estimate, which is extremely costly when p is large. Ruppert (1985) assumes that direct measurements of the gradient $\boldsymbol{g}(\cdot)$ are available, as in the stochastic gradient setting of Chapter 5. He then forms a Hessian estimate by taking a finite difference of gradient measurements. Hence, $O(p)$ measurements of $\boldsymbol{g}(\cdot)$ are

required for each update step in estimating θ. A type of second-order optimal convergence for SA is reported in Ruppert (1991), Polyak and Juditsky (1992), and Dippon and Renz (1997) based on the idea of iterate averaging. This was briefly discussed in Section 7.7.

The algorithm here is in the spirit of adaptive (matrix) gain SA algorithms such as those considered in Benveniste et al. (1990, Chaps. 3 and 4) in that a matrix gain is estimated concurrently with an estimate of the parameters of interest. It differs, however, in the relative lack of prior information required (especially in the gradient-free case) and in the small number of loss and/or gradient measurements needed per iteration. Because the algorithm is a stochastic analogue of the Newton–Raphson method, the algorithm provides a measure of transform invariance, as discussed in Section 1.4. That is, the method automatically scales the θ updates when there are significant differences in the magnitudes of the elements in θ.

The ASP approach is composed of two parallel recursions: one for θ and one for the Hessian of $L(\theta)$. The two core recursions are, respectively,

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \bar{\bar{H}}_k^{-1} G_k(\hat{\theta}_k), \quad \bar{\bar{H}}_k = f_k(\bar{H}_k), \tag{7.13a}$$

$$\bar{H}_k = \frac{k}{k+1}\bar{H}_{k-1} + \frac{1}{k+1}\hat{H}_k, \quad k = 0, 1, 2, \ldots, \tag{7.13b}$$

where a_k is a nonnegative scalar gain coefficient, $G_k(\hat{\theta}_k)$ is the input information related to $g(\hat{\theta}_k)$ (i.e., the gradient approximation $\hat{g}_k(\hat{\theta}_k)$ from (7.2) in the gradient-free case or the direct observation $\partial Q/\partial\theta$ in the stochastic gradient case of Chapter 5), f_k: $\mathbb{R}^{p\times p} \to$ {positive definite $p\times p$ matrices} is a mapping designed to cope with possible nonpositive-definiteness of $\bar{H}_k$, and $\hat{H}_k$ is a per-iteration estimate of the Hessian, $H = H(\theta)$, discussed below. (Note that at $k = 0$ in (7.13b), $\bar{H}_{k-1} = \bar{H}_{-1}$ is unspecified—and irrelevant—since the multiplier $k/(k+1) = 0$.)

Eqn. (7.13a) is a stochastic analogue of the Newton–Raphson algorithm. Eqn. (7.13b) is simply a recursive calculation of the sample mean of the per-iteration Hessian estimates.[2] Initialization of the two recursions is discussed in Subsection 7.8.2. Because $G_k(\hat{\theta}_k)$ has a known form, the parallel recursions in (7.13a, b) can be implemented once $\hat{H}_k$ is specified. The remainder of this section focuses on two specific implementations of the ASP approach above: 2SPSA (second-order SPSA) for applications in the gradient-free case and 2SG (second-order stochastic gradient) for applications in the gradient-based case (as in Chapter 5).

[2] It is also possible to use a weighted average or sliding window method (where only the most recent $\hat{H}_k$ values are used in the recursion) to determine $\bar{H}_k$. Formal convergence of $\bar{H}_k$ may still hold under such weighting provided that the analogue to (A.10) and (A.13) in the proof of Theorem 2a in Spall (2000) holds.

We now present the per-iteration Hessian estimate $\hat{\boldsymbol{H}}_k$. As with the basic first-order SPSA algorithm, let c_k be a positive scalar (decaying to zero for formal convergence) and $\boldsymbol{\Delta}_k \in \mathbb{R}^p$ be a user-generated mean-zero random vector; conditions on c_k, $\boldsymbol{\Delta}_k$, and other relevant quantities are given in Spall (2000). These conditions are close to those of basic SPSA (e.g., $\boldsymbol{\Delta}_k$ being a vector of independent Bernoulli ± 1 random variables satisfies these conditions, but a vector of uniformly or normally distributed random variables does not). Examples of valid gain sequences are given below. The formula for estimating the Hessian at each iteration is

$$\hat{\boldsymbol{H}}_k = \frac{1}{2}\left\{\frac{\delta \boldsymbol{G}_k}{2c_k}\left[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \ldots, \Delta_{kp}^{-1}\right] + \left(\frac{\delta \boldsymbol{G}_k}{2c_k}\left[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \ldots, \Delta_{kp}^{-1}\right]\right)^T\right\}, \quad (7.14)$$

where

$$\delta \boldsymbol{G}_k = \boldsymbol{G}_k^{(1)}(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k) - \boldsymbol{G}_k^{(1)}(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k),$$

and $\boldsymbol{G}_k^{(1)}$ may or may not equal $\boldsymbol{G}_k$, depending on the setting. (The form in (7.14) is equivalent to the "vector divide" form in Spall, 2000.) In particular, for 2SPSA, there are advantages to using a *one-sided* gradient approximation in order to reduce the total number of function evaluations (vs. the standard two-sided form in (7.2)), while for 2SG, usually $\boldsymbol{G}_k^{(1)} = \boldsymbol{G}_k = \partial Q/\partial \boldsymbol{\theta}$, as in Chapter 5.

Note that all elements of $\hat{\boldsymbol{\theta}}_k$ are varied simultaneously (and randomly) in forming $\hat{\boldsymbol{H}}_k$, as opposed to the finite-difference forms in, for example, Fabian (1971) and Ruppert (1985), where the elements of $\boldsymbol{\theta}$ are changed deterministically one at a time. The symmetrizing operation in (7.14) (the multiple 1/2 and the indicated sum) is convenient in the optimization case being emphasized here to maintain a symmetric Hessian estimate in finite samples. In the general root-finding case, where $\boldsymbol{H}(\boldsymbol{\theta})$ represents a Jacobian matrix, the symmetrizing operation should not be used when the Jacobian is not necessarily symmetric.

For 2SPSA, the core gradient approximation $\boldsymbol{G}_k(\hat{\boldsymbol{\theta}}_k)$ is taken as $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ in eqn. (7.2), requiring two measurements of L, $y(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k)$ and $y(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k)$. In addition to this gradient approximation, these two measurements are employed toward generating the one-sided gradient approximations $\boldsymbol{G}_k^{(1)}(\hat{\boldsymbol{\theta}}_k \pm c_k\boldsymbol{\Delta}_k)$ used in forming $\hat{\boldsymbol{H}}_k$. Two additional measurements $y(\hat{\boldsymbol{\theta}}_k \pm c_k\boldsymbol{\Delta}_k + \tilde{c}_k\tilde{\boldsymbol{\Delta}}_k)$ are used in generating the one-sided approximations as follows:

$$\boldsymbol{G}_k^{(1)}(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k)=\frac{y(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k+\tilde{c}_k\tilde{\Delta}_k)-y(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k)}{\tilde{c}_k}\begin{bmatrix}\tilde{\Delta}_{k1}^{-1}\\ \tilde{\Delta}_{k2}^{-1}\\ \vdots\\ \tilde{\Delta}_{kp}^{-1}\end{bmatrix}, \quad (7.15)$$

with $\tilde{\Delta}_k=[\tilde{\Delta}_{k1},\tilde{\Delta}_{k2},....,\tilde{\Delta}_{kp}]^T$ generated in the same statistical manner as Δ_k, but independently of Δ_k (in particular, choosing $\tilde{\Delta}_{ki}$ as independent Bernoulli ± 1 random variables is a valid—but not necessary—choice), and with $\tilde{c}_k$ satisfying conditions similar to c_k.

Let us summarize some examples of gains that satisfy the conditions in Spall (2000) for convergence and asymptotic normality of 2SPSA and 2SG. For both implementations, we can take a_k and c_k in the form given in Theorem 7.2 of Section 7.4. For 2SPSA, we also have $\tilde{c}_k = \tilde{c}/(k+1)^{\gamma}$, $\tilde{c} > 0$. With these gain forms, examples of specific coefficient values for 2SPSA are: $\alpha = 0.602$, $\gamma = 0.101$ or $\alpha = 1, \gamma = 1/6$. For 2SG, $1/2 < \alpha \leq 1$ is valid together with $0 < \gamma < 1/2$.

To illuminate the underlying simplicity of ASP, let us now provide some informal motivation for the $\hat{\boldsymbol{H}}_k$ form in eqn. (7.14). The arguments below are formalized in the theorems of Spall (2000). Suppose that L is four-times continuously differentiable in a neighborhood of $\hat{\boldsymbol{\theta}}_k$. Then, simple Taylor series arguments show that

$$\begin{aligned}E(\delta\boldsymbol{G}_k \mid \hat{\boldsymbol{\theta}}_k,\Delta_k)&=\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k+c_k\Delta_k)-\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k-c_k\Delta_k)+O(c_k^3)\\ &\equiv \delta\boldsymbol{g}_k+O(c_k^3) \quad (O(c_k^3)=\boldsymbol{0} \text{ in the 2SG case}),\end{aligned} \quad (7.16)$$

where this result is immediate in the 2SG case and follows as in Spall (1992, Lemma 1) by a Taylor series argument in the 2SPSA case (where the $O(c_k^3)$ term is the difference of the two $O(c_k^2)$ bias terms in the one-sided SP gradient approximations; see Exercise 7.13). Let δG_{ki} and δg_{ki} be the ith components of $\delta\boldsymbol{G}_k$ and $\delta\boldsymbol{g}_k$. By an expansion of each of $\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k \pm c_k\Delta_k)$ for any i, j,

$$\begin{aligned}E\left(\frac{\delta G_{ki}}{2c_k\Delta_{kj}}\middle|\hat{\boldsymbol{\theta}}_k,\Delta_k\right)&=E\left(\frac{\delta g_{ki}}{2c_k\Delta_{kj}}\middle|\hat{\boldsymbol{\theta}}_k,\Delta_k\right)+O(c_k^2)\\ &=H_{ij}(\hat{\boldsymbol{\theta}}_k)+\sum_{\ell\neq j}H_{i\ell}(\hat{\boldsymbol{\theta}}_k)\frac{\Delta_{k\ell}}{\Delta_{kj}}+O(c_k^2),\end{aligned}$$

where H_{ij} denotes the ijth component of $\boldsymbol{H}$ and the $O(c_k^2)$ term in the second line absorbs higher-order terms in the expansion of δg_{ki}. Then, since $E(\Delta_{k\ell}/\Delta_{kj}) = 0$ for all $j \neq \ell$ by the assumptions for Δ_k,

$$E\left(\frac{\delta G_{ki}}{2c_k \Delta_{kj}} \middle| \hat{\theta}_k \right) = H_{ij}(\hat{\theta}_k) + O(c_k^2),$$

implying that the Hessian estimate $\hat{H}_k$ is nearly unbiased with the bias disappearing at rate $O(c_k^2)$. The addition operation in (7.14) simply forces the per-iteration estimate to be symmetric.

7.8.2 Implementation Aspects of Adaptive SPSA

The two recursions in (7.13a,b) are the foundation for the ASP approach. However, as is typical in all stochastic algorithms, the specific implementation details are important. Eqns. (7.13a,b) do not fully define these details. The five points below have been found important in making ASP perform well in practice. More complete guidelines are given in Spall (2000).

A. **θ and H initialization.** Typically, eqn. (7.13a) is initialized at some $\hat{\theta}_0$ believed to be near θ^*. One may wish to run standard first-order SA (i.e., (7.13a) without $\bar{\bar{H}}_k^{-1}$) or some other "rough" optimization approach (e.g., the random search methods of Chapter 2) for some period to move the initial θ for ASP closer to θ^*. The user has the option of initializing or not initializing the H recursion with prior information. If prior information is available, the recursion may be initialized at some value, say, $\bar{H}_0 = \rho I_{p\times p}$, $\rho \geq 0$, or some other positive semidefinite matrix reflecting available prior information (e.g., if one knows that the θ elements will have very different magnitudes, then $\bar{H}_0$ may be chosen to approximately scale for the differences). Without initializing based on prior information, $\bar{H}_0$ may be computed directly as $\hat{H}_0$.

B. **Numerical issues and choice of $\bar{\bar{H}}_k$.** Since $\bar{H}_k$ may not be positive definite, especially for small k (even if $\bar{H}_0$ is positive definite), it is generally recommended that $\bar{H}_k$ in (7.13b) not be used directly in (7.13a). Hence, as shown in (7.13a), it is recommended that $\bar{H}_k$ be replaced by another matrix $\bar{\bar{H}}_k$ that is closely related to $\bar{H}_k$. Spall (2000) discusses ways in which $\bar{H}_k$ can be transformed to obtain $\bar{\bar{H}}_k$.

C. **Gradient/Hessian averaging.** At each iteration, it may be desirable to average several $\hat{H}_k$ and $G_k(\hat{\theta}_k)$ values despite the additional cost. This may be especially true in a high-noise environment.

D. **Gain selection.** The principles outlined in Section 7.5 are useful here as well for practical selection of the gains a_k, c_k, and, in the 2SPSA case, $\tilde{c}_k$. For 2SPSA and 2SG, the critical gain a_k can simply be picked as $1/(k+1)$ to achieve asymptotic near-optimality or optimality, respectively, although this may not be ideal in practical finite-sample problems (a slower decay is usually preferred in practice). In the 2SPSA

case, it may be desirable to choose $\tilde{c}_k$ larger than c_k to enhance numerical stability.

E. **Blocking.** At each iteration, block "bad" steps if the new estimate for θ fails a certain criterion (i.e., set $\hat{\theta}_{k+1} = \hat{\theta}_k$ in going from k to $k+1$). $\bar{H}_k$ should typically continue to be updated even if $\hat{\theta}_{k+1}$ is blocked. The most obvious blocking applies when θ must satisfy constraints; an updated value may be blocked or modified if a constraint is violated. There are two methods (say, E.1 and E.2) that one might use to implement blocking when constraints are not the limiting factor, with E.1 based on $\hat{\theta}_k$ and $\hat{\theta}_{k+1}$ directly and E.2 based on loss measurements. Both of E.1 and E.2 may be implemented in a given application. In E.1, the step from $\hat{\theta}_k$ to $\hat{\theta}_{k+1}$ is blocked if $\|\hat{\theta}_{k+1} - \hat{\theta}_k\| > \text{tol}_\theta$, where the norm is any convenient distance measure and $\text{tol}_\theta > 0$ is some reasonable maximum distance (tolerance) to cover in one step. The rationale behind E.1 is that a well-behaving algorithm should be moving toward the solution in a smooth manner and very large steps are indicative of potential divergence. The second method, E.2, is based on blocking the step if $y(\hat{\theta}_{k+1})$ (or an average of y values) is not sufficiently near (or lower) than $y(\hat{\theta}_k)$. In a setting where the noise in the loss measurements tends to be large (say, larger than the allowable difference between $L(\theta^*)$ and $L(\hat{\theta}_{\text{final}})$), it is generally undesirable to use E.2 due to the large number of loss measurements that must be averaged to obtain meaningful information about the relative old and new loss values.

Let us close this subsection with a few summary comments about the implementation aspects above. Without the second blocking procedure (E.2) in use, 2SPSA requires *four* measurements y per iteration, *regardless* of the dimension p: two for the standard $\boldsymbol{G}_k(\cdot) = \hat{\boldsymbol{g}}_k(\cdot)$ estimate and two new values for the one-sided SP gradients $\boldsymbol{G}_k^{(1)}(\cdot)$. For 2SG, *three* gradient measurements $\boldsymbol{G}_k(\cdot) = \partial Q/\partial\theta$ are needed, again independent of p. If the second blocking procedure (E.2) is used, one or more additional y measurements are needed for both 2SPSA and 2SG. The use of gradient/Hessian averaging (C) also increases the number of loss or gradient evaluations. E.1 may be used anytime while E.2 is more appropriate in a low- or no-noise setting. While E.1 helps to prevent divergence, it lacks direct insight into whether the loss function is improving. E.2 does provide that insight but requires additional y measurements, the number of which might grow prohibitive in a high-noise setting.

7.8.3 Theory on Convergence and Efficiency of Adaptive SPSA

Spall (2000) presents asymptotic theory showing the a.s. convergence of $\hat{\theta}_k$ and $\bar{H}_k$ to θ^* and $\boldsymbol{H}(\theta^*)$, respectively, in both the 2SPSA and 2SG settings. Further, conditions are shown for the asymptotic normality of the standardized

quantity $k^{\beta/2}(\hat{\theta}_k - \theta^*)$, $\beta > 0$. This normality is then used to analyze the limiting efficiency of the general ASP approach. Let μ and Σ be the mean vector and covariance matrix in the asymptotic distribution for $k^{\beta/2}(\hat{\theta}_k - \theta^*)$. Then the large-sample root-mean-squared (RMS) error of $\hat{\theta}_k$ based on the asymptotic distribution is $\sqrt{[\mu^T\mu + \text{trace}(\Sigma)]/k^{\beta}}$. As in the standard SPSA and stochastic gradient algorithms, the best (greatest) β is $\beta = 2/3$ for the 2SPSA case and $\beta = 1$ for the 2SG case (both follow by setting $\alpha = 1$ in the standard gain form $a_k = a/(k+1+A)^{\alpha}$).

To characterize the asymptotic efficiency results based on the asymptotic distributions, let RMS^*_{SPSA} and RMS^*_{SG} represent the *best possible* RMS errors of the normalized $\hat{\theta}_k$ when using the basic SPSA (gradient-free) and stochastic gradient approaches. The best possible RMS errors require gain sequences based on exact information on the third derivative of L (basic SPSA) and the second derivatives of L (stochastic gradient) (Dippon and Renz, 1997). This information, of course, is generally unavailable. Hence, RMS^*_{SPSA} and RMS^*_{SG} represent ideal values that are usually unavailable in practice. (As mentioned in Section 4.4, RMS^*_{SG} is derived from the inverse Fisher information matrix; this matrix is discussed in detail in Section 13.3.) Letting RMS_{2SPSA} and RMS_{2SG} denote the corresponding large-sample RMS errors for the normalized 2SPSA and 2SG estimates when $a_k = 1/(k+1)$ (i.e., $\sqrt{[\mu^T\mu + \text{trace}(\Sigma)]/k^{\beta}}$ from above), we find

$$\frac{RMS_{\text{2SPSA}}}{RMS^*_{\text{SPSA}}} < 2 \quad \text{and} \quad \frac{RMS_{\text{2SG}}}{RMS^*_{\text{SG}}} = 1. \tag{7.17}$$

The interpretation of (7.17) is that for the SPSA setting, the 2SPSA algorithm produces an estimate with an asymptotic RMS error *less than* twice the error from the best possible (infeasible) algorithm (requiring the above-mentioned third derivative knowledge). For the stochastic gradient setting, the 2SG algorithm produces an error that is asymptotically *equal* to the best possible. Numerical studies in Spall (2000) show the power of the 2SPSA and 2SG approaches. Luman (2000) applies 2SPSA in a simulation-based optimization problem and demonstrates the improvement possible over basic SPSA when there are very different magnitudes for the elements in θ (i.e., an illustration of the above-mentioned transform invariance property).

While the ASP approach (via the 2SPSA and 2SG implementations) can be very powerful, there are no guarantees that the small-sample performance will be superior to the basic SPSA or stochastic gradient algorithms. It may take many iterations to accumulate the information needed to produce a credible Hessian estimate, especially in the 2SPSA case where only noisy loss measurements are available. (Zhu and Spall, 2002, present a modification to cope with finite-sample concerns.) Further, as with the deterministic Newton–Raphson algorithm, ASP may be more "brittle" in the sense of being

less robust to violations of the formal conditions for convergence. Nevertheless, one of the most important practical implications of (7.17) is that for $\hat{\boldsymbol{\theta}}_0$ sufficiently close to $\boldsymbol{\theta}^*$, 2SPSA or 2SG (as appropriate) will yield a good (or best for 2SG) large-sample solution based on the simple choice of $a_k = 1/(k+1)$. This helps alleviate the potentially nettlesome issue of gain selection subject to some of the practical tips in Subsection 7.8.2.

7.9 CONCLUDING REMARKS

This chapter has described the simultaneous perturbation stochastic approximation approach for stochastic search and optimization. The SPSA method rests on the idea of changing all the parameters in the problem simultaneously to construct gradient estimates. This contrasts with conventional one-at-a-time changes (such as the finite-difference method of Chapter 6). In high-dimensional problems of practical interest, such simultaneous changes admit an efficient implementation by greatly reducing the number of loss function evaluations required to carry out the optimization process.

As with other SA approaches, SPSA is explicitly designed to cope with noisy measurements of the loss function, although it is frequently applied in problems with noise-free measurements. Aside from differences in the mechanics of the algorithms, the formal justification for handling noisy measurements distinguishes SPSA from the random search methods of Chapter 2 and the simulated annealing and evolutionary computation methods of Chapters 8–10.

While the basic method applies in "smooth," unconstrained problems with no extraneous local minima, there are numerous extensions to the algorithm and theory to accommodate other cases. In particular, extensions exist for discrete optimization, constrained problems, and global optimization with multiple local minima. Interestingly, under some conditions, the *basic* SPSA algorithm is guaranteed to converge to a global solution. This appealing property implies that the algorithm automatically combines the broad search needed for global optimization with the localized search that is tied to gradient information (because the algorithm produces gradient information from the loss measurements). There is no need to explicitly transition from a "rough" global optimizer to a refined local optimizer as is common in practice.

Aside from the first-order SPSA algorithm, this chapter summarized a second-order SA approach that is a stochastic analogue of the Newton–Raphson method. This approach applies in either the standard SPSA setting, where only noisy loss function evaluations are available, or in the stochastic gradient-based/root-finding (Robbins–Monro) setting, where noisy gradient evaluations are available. The algorithm produces an asymptotically near-optimal or optimal root-mean-squared error for the iterate. This adaptive simultaneous perturbation algorithm is based on a relatively simple method for estimating the Hessian

matrix of the loss function L. The Hessian estimation capability can also be used for computing the Fisher information matrix (see Section 13.3), a quantity important in parameter estimation, model selection, and experimental design.

Further general information and references on theory and applications of SPSA are available at *www.jhuapl.edu/SPSA* (also available through the book's Web site).

7.10 APPENDIX: CONDITIONS FOR ASYMPTOTIC NORMALITY

The conditions below are used in the asymptotic normality result of Section 7.4 (together with conditions B.1″–B.6″ stated in Section 7.3).

B.7″ The continuity and equicontinuity assumptions about $E[(\varepsilon_k^{(+)}-\varepsilon_k^{(-)})^2|\Im_k]$ in Spall (1992, Prop. 2) hold. (These assumptions are automatically satisfied if the $\varepsilon_k^{(\pm)}$ are independent of $\Im_k$; equicontinuity is a stricter form of continuity.)

B.8″ $\boldsymbol{H}(\boldsymbol{\theta}^*)$ is positive definite where $\boldsymbol{H}(\boldsymbol{\theta})$ is the Hessian matrix of $L(\boldsymbol{\theta})$. Further, let λ_i denote the ith eigenvalue of $a\boldsymbol{H}(\boldsymbol{\theta}^*)$ (the a here is the a in a_k). If $\alpha = 1$, then $\beta < 2\min_i(\lambda_i)$.

B.9″ $E(\Delta_{ki}^2) \to \rho$, $E(\Delta_{ki}^{-2}) \to \rho'$, and $E[(\varepsilon_k^{(+)}-\varepsilon_k^{(-)})^2|\Im_k] \to \rho''$ for strictly positive constants ρ, ρ', and ρ'' (a.s. in the latter case) as $k \to \infty$ (often, $E(\Delta_{ki}^2)$ and $E(\Delta_{ki}^{-2})$ will be *equal* to ρ and ρ', respectively).

EXERCISES

7.1 Consider the quadratic function $L(\boldsymbol{\theta}) = 2t_1^2 + t_2^2$, $\boldsymbol{\theta} = [t_1, t_2]^T$, and suppose that the measurements of $L(\boldsymbol{\theta})$ have no noise (i.e., $y(\boldsymbol{\theta}) = L(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$). Perform the following tasks (pencil-and-paper calculations will work fine):

(a) From an initial condition $[1, 1]^T$, with $a_0 = 0.1$, $c_0 = 1.0$, calculate what $\hat{\boldsymbol{\theta}}_1$ will be for all possible combinations of $\boldsymbol{\Delta}_0$ with each component of $\boldsymbol{\Delta}_0$ distributed as Bernoulli ± 1.

(b) Using the $L(\boldsymbol{\theta})$ values as the measure of performance, show that $\hat{\boldsymbol{\theta}}_1$ is guaranteed to be an improvement over $\hat{\boldsymbol{\theta}}_0$.

(c) Show by direct calculation that $\hat{\boldsymbol{g}}_0(\hat{\boldsymbol{\theta}}_0)$ is an unbiased estimator of $\boldsymbol{g}(\hat{\boldsymbol{\theta}}_0)$. (This lack of bias is a consequence of the quadratic $L(\boldsymbol{\theta})$.)

7.2 For $E(\Delta_{ki}/\Delta_{km})$ to exist in (7.3), show that the inverse moment $E(|1/\Delta_{km}|)$ must be finite when $E(\Delta_{ki}) = 0$.

7.3 For the skewed-quartic loss function in Example 6.6 (Section 6.7), compare SPSA with a valid perturbation distribution and two invalid perturbation distributions. In the valid case, let the perturbations (Δ_{ki}) be i.i.d. Bernoulli ± 1 and in the invalid cases, let the perturbations be i.i.d. uniform over

$[-\sqrt{3}, \sqrt{3}]$ and i.i.d. $N(0, 1)$. (Note that the valid and invalid perturbation distributions all have mean 0 and variance 1.) Let $p = 10$ and the gains be in the standard form $a_k = a/(k+1+A)^{\alpha}$ and $c_k = c/(k+1)^{\gamma}$. Suppose that the noise ε in the measurements $y(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \varepsilon$ is i.i.d. $N(0, 0.1^2)$, $\hat{\boldsymbol{\theta}}_0 = [1, 1, \ldots, 1]^T$ (so $L(\hat{\boldsymbol{\theta}}_0) = 4.178$), and the gains satisfy $a = 0.5$, $A = 50$, $c = 0.1$, $\alpha = 0.602$, and $\gamma = 0.101$ (chosen via the gain-selection guidelines in Section 7.5). With no constraints imposed on $\boldsymbol{\theta}$, compare single replications of 2000 measurements for the valid and invalid implementations and comment on the relative performance of the implementations.

7.4 Consider the setting of Exercise 7.3 for the Bernoulli (valid) perturbations *with the exception* of imposing the hypercube constraint $\hat{\boldsymbol{\theta}}_k \in \Theta = [-1, 1]^{10}$ for all k. (The perturbed values $\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k$ should not be constrained.) Use the appropriate two-sample t-test (Appendix B) to compare the terminal loss values from 20 independent replications with 2000 measurements in the unconstrained and constrained cases. (This exercise demonstrates the advantage of imposing constraints if possible.)

7.5 Prove (7.6) under the stated conditions in Section 7.2. (Hint: Recall from integration theory: $\int_D |f_1(x) f_2(x)| dx \leq \sup_{x \in D} |f_1(x)| \int_D |f_2(x)| dx$ for two functions f_1 and f_2 and some domain of integration D.)

7.6 Prove inequality (7.7) and give two specific examples of exponent terms η and τ.

7.7 Consider the problem of minimizing $L(\boldsymbol{\theta}) = t_1^4 + t_1^2 + t_1 t_2 + t_2^2$, $\boldsymbol{\theta} = [t_1, t_2]^T$, where ε is i.i.d. $N(0, 1)$ noise. Define the segmented uniform distribution as in Figure 7.2, where the probability is evenly split that the random variable lies in either the upper interval (u, v) or the lower interval $(-v, -u)$, $v > u > 0$. Let $\hat{\boldsymbol{\theta}}_0 = [1, 1]^T$ and let each experiment below entail 100 replications. Use the gains $a_k = a/(k+1+A)^{0.602}$ and $c_k = c/(k+1)^{0.101}$ as in step 0 in Subsection 7.5.1.

(a) Compute the sample mean of the terminal loss values after 1000 iterations of SPSA when the Δ_{ki} are distributed Bernoulli ± 1 for all k, i. Use the coefficient values $a = 0.05$, $A = 100$, and $c = 1.0$. Next compute the same when the Δ_{ki} have a segmented uniform distribution with $u = 0.4091$ and $v = 1.4909$ (the strange values for u and v are so that the variance is unity, the same as the Bernoulli distribution). Use the coefficient values $a = 0.13$, $A = 100$, and $c = 0.55$. (Each of the two sets of gain coefficients is approximately optimized for the respective SPSA implementation.) Determine a P-value for comparing the two sample means.

(b) Perform the same analysis as in part (a) but with only 10 iterations of SPSA. Use the same gain coefficient values as in part (a), except that $A = 1$. Determine a P-value for comparing the two sample means.

(c) Offer an explanation for the conclusions you reach in parts (a) and (b).

7.8 Create a specific U-shaped density that satisfies (7.8) for an arbitrarily small $\tau > 0$.

Note: Use Bernoulli ±1 perturbations for the components of Δ_k in the numerical studies among the exercises below.

7.9 Given a desirable step size of 0.1 for all elements of θ, use the semiautomatic method of Subsection 7.5.2 to find reasonable gains for SPSA as applied to the loss function $L(\theta) = (\theta^T\theta/5 - 25)^2$, with $p = 10$. Suppose that 1000 noisy loss measurements are used in the search process and $\hat{\theta}_0 = [6, 6, \ldots, 6]^T$.

(a) Find the values of c, A, and a when ε is i.i.d. $N(0, 2^2)$.

(b) Find the values of c, A, and a when $\varepsilon = \varepsilon(\theta) = [\theta^T, 1]V$ and V is an i.i.d. vector with distribution $N(\mathbf{0}, 0.1^2\boldsymbol{I}_{11})$.

(c) Run one replication of SPSA for the noise and gain combination in part (a) and one replication for part (b). How do the loss values at each of the two terminal θ values compare with $L(\hat{\theta}_0)$ and $L(\theta^*)$ for any $\theta^* \in \Theta^*$? (θ^* is not unique.)

7.10 Consider the loss function and noise of Exercise 7.9(a):

(a) Use basic SPSA with semiautomatic gains as found in Exercise 7.9(a). Determine the mean terminal loss value and approximate 95 percent confidence interval (using the standard t-distribution approach of Appendix B) based on 40 independent replications of 1000 noisy loss measurements.

(b) Repeat part (a) except for using gains that are manually "tuned" to the problem. Compare the results with the results in part (a).

7.11 There is no unique minimum for the loss function in Exercise 7.9.

(a) Identify Θ^*.

(b) Identify two distinct initial conditions having the same distance to the nearest point in Θ^* as $\hat{\theta}_0 = [6, 6, \ldots, 6]^T$ does to its nearest point.

(c) Run one replication of SPSA for each of the three initial conditions in part (b) (the two new ones plus the original one used in Exercise 7.9). Use 1000 loss measurements with noise as in Exercise 7.9(a) and use the same gains a_k and c_k in the three runs (you may use gains as in Exercise 7.9(a)). How do the three terminal loss values compare and how do the corresponding terminal θ estimates relate to the initial conditions and to Θ^*?

7.12 Based on the conditions given for (7.5) and (7.6) for the two-measurement gradient approximation, show that the one-measurement gradient approximation in (7.12) has an $O(c_k^2)$ bias (same order as the standard two-measurement form).

7.13 Establish the $O(c_k^3)$ bias in (7.16) when $\boldsymbol{G}_k^{(1)}$ is the one-sided SP gradient approximation in (7.15). (Hint: First, write down an expression for the $O(c_k^2)$ bias in $\boldsymbol{G}_k^{(1)}(\theta)$ using arguments such as in Section 7.2. Then, analyze the difference of the two biases for each of $\boldsymbol{G}_k^{(1)}(\hat{\theta}_k \pm c_k\Delta_k)$.)

7.14 Consider the loss function

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{10} i t_i + \prod_{i=1}^{10} t_i^{-1}, \; t_i > 0 \text{ for all } i,$$

where $\boldsymbol{\theta} = [t_1, t_2, \ldots, t_{10}]^T$. Suppose that $\varepsilon = \varepsilon(\boldsymbol{\theta}) = [\boldsymbol{\theta}^T, 1]\boldsymbol{V}$, $\boldsymbol{V}$ is i.i.d. $N(\mathbf{0}, 0.001^2 \boldsymbol{I}_{11})$, and that $\hat{\boldsymbol{\theta}}_0 = 1.1 \times \boldsymbol{\theta}^*$. Carry out the following tasks:

(a) Determine $\boldsymbol{\theta}^*$ by whatever means desired (a deterministic problem since $L(\boldsymbol{\theta})$ is known).

(b) Run basic SPSA with the noise included in the loss measurements. Use the standard gain form and $a = 0.01$, $A = 1000$, $c = 0.015$, $\alpha = 0.602$, and $\gamma = 0.101$ (these gains reflect tuning to achieve good performance). Determine the mean terminal loss value and approximate 90 percent confidence interval (using the standard t-distribution approach of Appendix B) based on 10 replications of 20,000 noisy loss measurements and $\Theta = (0, \infty)^{10}$. Comment on the performance relative to $\hat{\boldsymbol{\theta}}_0$ and $\boldsymbol{\theta}^*$ (e.g., use a measure such as $L_{\text{norm}}(\boldsymbol{\theta})$ in Example 7.3).

(c) For the noisy case of (b), run the 2SPSA approach with a naïve gain $a_k = 1/(k+1)$. Use c_k as in part (b), $\tilde{c} = 2c$, and $\text{tol}_\theta = 0.2$. Initialize the Hessian estimate as $\bar{\boldsymbol{H}}_0 = 500\boldsymbol{I}_{10}$. Report results as in part (b) and compare with the performance of the tuned SPSA algorithm in part (b). Comment on the performance relative to $\hat{\boldsymbol{\theta}}_0$ and $\boldsymbol{\theta}^*$.