# General Bounds and Finite-Time Improvement for Stochastic Approximation Algorithms*

Mark Broadie† Columbia University

Deniz M. Cicek‡ Columbia University

Assaf Zeevi§ Columbia University

February 3, 2009

**Abstract**

We consider the Kiefer-Wolfowitz (KW) stochastic approximation algorithm and derive general upper bounds on its mean-squared error. The bounds are established using an elementary induction argument and phrased directly in the terms of tuning sequences of the algorithm. From this we deduce the non-necessity of one of the main assumptions imposed on the tuning sequences in the Kiefer-Wolfowitz paper and essentially all subsequent literature. The optimal choice of sequences is derived for various cases of interest, and an adaptive version of the KW algorithm is proposed with the aim of improving its finite-time behavior. The key idea is to dynamically scale and shift the tuning sequences to better match them with characteristics of the unknown function and noise level, and thus improve algorithm performance. Numerical results are provided which illustrate that the proposed algorithm retains the convergence properties of the original KW algorithm while dramatically improving its performance in some cases.

## 1 Introduction

**Background and motivation.** The term stochastic approximation refers to a broad class of optimization problems in which function values can only be computed in the presence of noise. Representative examples include stochastic estimation of a zero crossing, first introduced in the work of Robbins and Monro [18], and stochastic estimation of the point of maximum, first studied by Kiefer and Wolfowitz [13]. Such problems arise in a variety of fields including engineering, statistics, operations research and economics, and the literature on the topic is voluminous; cf. the survey paper by Lai [15] and the book by Kushner and Yin [14].

A natural setting in which one encounters the need for stochastic approximation algorithms is simulation-based optimization. Here it is only possible to evaluate a function by means of

*This work was supported by Credit Suisse and NSF Grant DMII-0447652.

†Graduate School of Business, e-mail: mnb2@columbia.edu

‡Graduate School of Business, email: dcicek05@gsb.columbia.edu (contact author)

§Graduate School of Business, e-mail: assaf@gsb.columbia.edu

simulation, and the observation noise is a direct consequence of the sample generating scheme; see, for example, [1, 2] for further discussion.

In this paper for concreteness we focus on the problem of sequential estimation of the point of maximum of an unknown function from noisy observations, noting that the main ideas developed in the paper extend in a straightforward manner to Robbins-Monro type algorithms; more specific commentary will be given in §2. In particular, we consider the following stochastic approximation scheme first studied by Kiefer and Wolfowitz [13]:

$$X_{n+1} = X_n + a_n \left( \frac{Y_{2n} - Y_{2n-1}}{c_n} \right), \quad n = 1, 2, \dots \tag{1}$$

Here $X_1$ is the initial condition (either deterministic or random), $\{a_n\}$ and $\{c_n\}$ are two real-valued, deterministic *tuning sequences*, and $Y_{2n}, Y_{2n-1}$ are drawn according to conditional distribution functions $H(y|X_n + c_n)$ and $H(y|X_n - c_n)$ which have uniformly bounded second moments. Assuming the regression function $f(x) := \int y dH(y|x)$ admits a unique point of maximum and is strongly concave, Kiefer and Wolfowitz [13] proved that the sequence $\{X_n\}$ generated by recursion (1) converges in probability to $x^*$, the unique maximizer of $f(\cdot)$, if $\{a_n\}$ and $\{c_n\}$ satisfy the following conditions:

(KW1) $c_n \to 0$ as $n \to \infty$;

(KW2) $\sum_{n=1}^{\infty} a_n = \infty$;

(KW3) $\sum_{n=1}^{\infty} \frac{a_n^2}{c_n^2} < \infty$;

(KW4) $\sum_{n=1}^{\infty} a_n c_n < \infty$.

Shortly after the publication of the KW algorithm, Blum [3] established that condition (KW4) is not necessary for convergence, leaving conditions (KW1)-(KW3) which have been imposed in almost all papers published on the subject since then. Roughly speaking, to have a convergent algorithm one requires that: i.) the gradient estimate localizes, hence $c_n$ should shrink to zero; ii.) the step-size sequence $a_n$ should shrink to zero, but in a manner that allows the algorithm to "cover" any distance from the initial point $X_1$ to the point of maximum, hence $\sum_n a_n$ diverges. If one adds the assumption that $a_n \to 0$ to (KW1) and (KW2), the role of (KW3) becomes questionable (and in fact, as this paper shows, superfluous).

A main focus in the literature has been establishing bounds on the mean-squared error (MSE) $\mathbb{E}|X_n - x^*|^2$, and deriving optimal rates at which the MSE converges to zero, under various assumptions on the unknown function and various modifications to the basic KW scheme; see, e.g., [7], [9], [10], [20]. A common thread in these papers is that they all rely on a key lemma by Chung [6] which restricts the tuning sequences $\{a_n\}$ and $\{c_n\}$ to be polynomial-like, specifically, of the form

$n^{-a}$ and $n^{-c}$, respectively, for some $a, c > 0$ such that conditions (KW1)-(KW3) hold. (Exceptions to this can be found in a stream of literature that develops weak convergence results; see, e.g., [5], [19], and more recently [16] as well as references therein.)

On a more practical side, the KW algorithm, notwithstanding the theoretical convergence guarantees, has often been seen to exhibit poor behavior in implementations. The main culprit seems to be the tuning sequences which may not match up well with the characteristics of the underlying function. Hence there is a need to *adapt* the choice of these sequences to observed data points. Among the first to tackle this issue was Kesten [12], who proposed a simple scheme to determine the step size at the $n$th iteration using the total number of sign changes of $\{X_m - X_{m-1} : m = 1, \ldots, n\}$. In a more recent paper, Andradóttir [2] observed divergence of the KW algorithm when applied to functions which are "too steep," and proposed to adjust for this using two independent gradient estimates at each iteration. A related issue arises when the magnitude of the step size is "too small" relative to the curvature of function, which may lead to a degraded rate of convergence. Work in [8], [11] and [17] point to this problem and propose some potential solutions; see also further discussion in §3.

The convergence theory and specification of tuning sequences subject to (KW1)-(KW3) hinges on the *global* strong concavity/convexity of the underlying function $f(\cdot)$; see conditions (F1) and (F2) in §2. This assumption is unrealistic when it comes to most application settings. Kiefer and Wolfowitz [13] identified this issue in their original paper, and proposed to "localize" the algorithm by restricting attention to a compact set (say, a closed bounded interval) which is known to contain the point of maximum. They argued that by projecting the iterates of the KW algorithm so that there will be no function evaluations outside of this set, one preserves the desired convergence properties without the need for the function to satisfy overly restrictive global regularity conditions. This *truncated* KW algorithm solves the divergence problem identified by Andradóttir [2], however it introduces the problem of *oscillatory behavior* of the iterates: if the magnitude of the step-size sequence ($\{a_n\}$) is chosen too large relative to the magnitude of the gradient, the algorithm may end up oscillating back and forth between the boundaries of the truncation interval (see further discussion in §3). Andradóttir [1] proposed an algorithm that adaptively determines the truncation interval, but still points to the oscillatory behavior as an open problem. Finally, poor performance is also observed when function evaluations tend to be "too noisy," degrading the quality of the gradient estimate. To the best of our knowledge there is no work in the literature addressing this issue.

**Main contributions.** This paper makes contributions along the two dimensions discussed above. On the theoretical end, we present a new induction-based approach to bounding the MSE of the KW algorithm. The proof is simple and yields general bounds that hold under broad assumptions on the tuning sequences; see Theorem 1. As a consequence of these bounds, we find

that assumption (KW3), which has been imposed in most of the literature, is in fact not necessary for the MSE to converge to zero (see §2.2.2). From these bounds we also deduce the optimal choice of the sequences $\{a_n\}$ and $\{c_n\}$ for a variety of cases of interest. Unlike previous literature, we do not impose polynomial decay a priori, but rather show how this is *derived* from minimizing the order of our general MSE bounds (see Proposition 1 and Proposition 3). Other settings such as quadratic-like functions (see Proposition 2) and functions that satisfy further smoothness assumptions (see Theorem 2) are discussed as well. Our analysis focuses on the one-dimensional case, but we illustrate how the core ideas can be applied in a multidimensional setting; see Remark 5 and Corollary 1.

Building on qualitative insights and intuition gleaned from our proofs, we present an adaptive version of the KW algorithm and illustrate via several examples its improved finite-time behavior. The algorithm is based on adaptively *scaling* the magnitude of the tuning sequences values, as well as *shifting* the index set. In particular, the rate degradation stemming from a step size that is "too small" is addressed by adaptively scaling up the $\{a_n\}$ sequence by a multiplicative constant. The oscillatory behavior that is due to a "too large" step size is solved by adaptively shifting the index of the $\{a_n\}$ sequence. Finally, the issue related to "large" simulation/estimation error in function evaluations is addressed by adaptively scaling up the $\{c_n\}$ sequence values. The MATLAB implementation of the algorithm can be downloaded from *www.columbia.edu/∼mnb2/broadie/research.html*.

**Remainder of the paper.** Section 2 gives the main theoretical results, and discusses some implications, in particular, the non-necessity of assumption (KW3). Section 3 describes our proposed adaptive algorithm. All proofs are given in Appendix A and Appendix B discusses extensions to the multidimensional setting.

# 2 Performance Bounds and Their Implications

## 2.1 Bounds on the mean squared error

Consider the recursion (1) in the previous section. Throughout the paper, we assume that

$$\sigma^2 := \sup_{x \in \mathbb{R}} \mathrm{Var}[Y|X = x] < \infty.$$

**Example 1** *A common setting for stochastic approximation is where, conditioned on $x$, $Y_i = f(x) + \varepsilon_i$, where $\{\varepsilon_i\}$ is a sequence of independent and identically distributed (i.i.d.) random variables with mean zero and finite variance.*

For the function $f$ to be maximized, we assume that:

(F1) There exist finite positive constants $K_0$ and $K_1$ such that $K_0|x - x^*| \leq |f'(x)| \leq K_1|x - x^*|$ for all $x \in \mathbb{R}$, and

(F2) $f'(x)(x - x^*) < 0$ for all $x \in \mathbb{R} \setminus \{x^*\}$.

**Remark 1** *Assumptions (F1) and (F2) are identical to those found in most previous literature. Assumption (F1) imposes a linearly growing envelope on the gradient. In essence, it guarantees that the function does not have flat regions away from the point of maximum. Assumption (F2) requires the function to be increasing for $x < x^*$ and decreasing for $x > x^*$, i.e., it has a "well-separated" point of maximum.*

The tuning sequences to be used in the algorithm, $\{a_n\}$ and $\{c_n\}$, are assumed to be positive and bounded, and for some finite constants $A, \tau_1$ and $\tau_2$ satisfy:

(S1) $a_n/c_n^2 \leq (a_{n+1}/c_{n+1}^2)(1 + Aa_{n+1})$ for all $n \geq 1$.

(S2) $c_n^2 \leq c_{n+1}^2(1 + Aa_{n+1})$ for all $n \geq 1$.

(S3) $a_n \to 0$ as $n \to \infty$.

(S4) either (i) $c_n^4/a_n \leq \tau_1$ or (ii) $c_n^4/a_n \geq \tau_2$, for all $n \geq 1$.

**Remark 2** *The sequences $a_n = \theta_a/n^a$ and $c_n = \theta_c/n^c$ for $0 < a \leq 1$ and $c \geq 0$ satisfy (S1)-(S4), but unlike most of the literature referenced in §1, these assumptions do not constrain $\{a_n\}$ and $\{c_n\}$ to be polynomial-like. In particular, they allow for a much broader class of sequences, some simple examples being $a_n = \theta_a/n, c_n = \theta_c/\log(n)$ and $a_n = \log(\log(n+2))/n, c_n = \theta_c$ with $\theta_a$ and $\theta_c$ being finite positive constants. We also note that the assumption "for all $n \geq 1$" in (S1)-(S4) is made mainly for simplicity; with obvious changes it can be replaced by "for all $n$ sufficiently large."*

The following is the main result of this section.

**Theorem 1** *Let $\{X_n\}$ be generated by the Kiefer-Wolfowitz stochastic approximation recursion given in (1) using $\{a_n\}$ and $\{c_n\}$ satisfying (S1)-(S4) with $A < 4K_0$. Then under assumptions (F1) and (F2),*

$$\mathbb{E}(X_{n+1} - x^*)^2 \leq \begin{cases} C_1 a_n/c_n^2 & \text{if} \quad c_n^4 \leq \tau_1 a_n \\ C_2 c_n^2 & \text{if} \quad c_n^4 \geq \tau_2 a_n \end{cases} \tag{2}$$

*for all $n \geq 1$, where $C_1$ and $C_2$ are finite positive constants identified explicitly in (30) and (33), respectively.*

**Proof outline.** We only sketch the key ideas here. The full proof is given in Appendix A.1. First, using assumptions (F1) and (F2) we derive bounds on the finite difference approximation of the gradient; see (15) and (17). Second, using the KW recursion (1) we express $(X_{n+1} - x^*)^2$ as a

function of $X_n$. Then, after some algebra, taking expectations and using gradient bounds we get the real-number recursion:

$$b_{n+1} \leq (1 - 4a_n K_0 + 8K_1^2 a_n^2) b_n + 2K_1 a_n c_n \sqrt{b_n} + 2\frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2 a_n^2 c_n^2. \tag{3}$$

where $b_n := \mathbb{E}(X_n - x^*)^2$.

Now, since $a_n \to 0$ as $n \to \infty$, $(1 - 4a_n K_0 + 8K_1^2 a_n^2) < 1$ holds for all $n$ suitably large and we eventually have a contraction in recursion (3). This ensures convergence of the mean-squared error to zero as $n \to \infty$. To derive bounds on the MSE we use a straightforward induction argument where assumptions (S1)-(S4) are required for the induction step. We first use assumptions (S1) and (S2) along with the induction hypothesis to identify the higher order terms; these turn out to be either $C_1 a_n/c_n^2$ or $C_2 c_n^2$. Then, to finish the proof, we rely on (S3) to show that all remaining terms are of lower order. (This step involves the study of the behavior of a certain quadratic equation given in (28).) Expressions for the constants $C_1$ and $C_2$ are identified explicitly as part of this analysis. ■

**Remark 3** ***(Error bounds for the maximum)*** *Using a simple Taylor expansion and assumption (F1), we can derive from Theorem 1 upper bounds on $f(x^*) - \mathbb{E}f(X_n)$. Specifically, we have*

$$\begin{aligned} f(x^*) - f(X_n) &= |x^* - X_n| \cdot |f'(\xi_n)| \quad \textit{for some } \xi_n \in (x^*, X_n) \\ &\leq K_1 |x^* - X_n| \cdot |\xi_n - x^*| \\ &\leq K_1 (X_n - x^*)^2, \end{aligned}$$

*where the first inequality follows from (F1) and the second since $\xi_n \in (x^*, X_n)$. Taking expectations and applying Theorem 1 we get*

$$f(x^*) - \mathbb{E}(f(X_n)) \leq \begin{cases} K_1 C_1 a_n/c_n^2 & \text{if} \quad c_n^4 \leq \tau_1 a_n \\ K_1 C_2 c_n^2 & \text{if} \quad c_n^4 \geq \tau_2 a_n, \end{cases} \tag{4}$$

*where $C_1, C_2, \tau_1, \tau_2$ are defined in Theorem 1.*

**Remark 4** ***(Truncated KW algorithm)*** *Thereom 1 requires the assumptions (F1) and (F2) to hold globally, which can be quite restrictive. This issue is also addressed in Kiefer and Wolfowitz [13] where they argue that it suffices to have assumptions (F1) and (F2) hold only on a compact interval $I_0 = [l, u]$, that is known to contain the point of maximum for the asymptotic theory to be valid. They propose projecting iterate $n+1$ onto a "truncation interval" $I_{n+1} = [l + c_{n+1}, u - c_{n+1}]$ at step $n$ so that there will be no function evaluations outside the interval $I_0$ (we assume $c_n < (u-l)/2$ for all $n \geq 1$). Such truncated algorithms are commonly used in the literature; see [1] and [11] and*

*references therein for some examples.*

*Using the same notation of the recursion given in (1), the "truncated KW algorithm" uses the recursion*

$$X_{n+1} = \Pi_{I_{n+1}} \left( X_n + a_n \left( \frac{Y_{2n} - Y_{2n-1}}{c_n} \right) \right) \tag{5}$$

*where $\Pi_{I_{n+1}}(\cdot)$ denotes the Euclidean projection operator onto the truncation interval $I_{n+1} = [l + c_{n+1}, u - c_{n+1}]$. The results of Theorem 1 still hold for the truncated KW algorithm. The proof follows the same lines of the proof of Theorem 1 using the contraction property of the Euclidean projection operator.*[1]

**Remark 5** ***(Multidimensional extensions)*** *The result in Theorem 1, and the proof that supports it, can be easily extended to certain multidimensional versions of the KW algorithm (e.g., that of Blum [4]), with some obvious modifications to assumptions (F1) and (F2). For an illustration, see Corollary 1 in Appendix B.*

**Remark 6** ***(Extensions to root-finding problems)*** *Consider the setting described by Robbins and Monro in [18]. The problem is to find the unique root $x^*$ of $f(x) - \xi = 0$, where $f(x) := \int y dH(y|x)$, from sequential noisy observations of $f(\cdot)$. Robbins and Monro [18] consider the following stochastic approximation scheme:*

$$X_{n+1} = X_n + a_n(\xi - Y_n) \quad n = 1, 2, \ldots \tag{6}$$

*Here, $Y_n$ is drawn according to the conditional distribution function $H(y|X_n)$ and is assumed to have uniformly bounded variance. The function $f(x)$ is assumed to be differentiable and monotonically increasing with $0 < K_0 \leq f'(x) \leq K_1$ for all $x \in \mathbb{R}$ and for some finite positive constants $K_0$, $K_1$. For any step-size sequence $\{a_n\}$ that satisfies $a_n \leq a_{n+1}(1 + Aa_{n+1})$ for some positive constant $A$ such that $A < K_0$, one can easily show that*

$$\mathbb{E}(X_{n+1} - x^*)^2 \leq Ca_n, \quad \text{for all } n \geq 1, \tag{7}$$

[1]The proof of Theorem 1 for the truncated KW algorithm is identical until equation (19), where we now have

$$\begin{aligned} Z_{n+1} &:= (X_{n+1} - x^*)^2 \\ &= \left[ \Pi_{I_{n+1}} \left( X_n + a_n \left( \frac{Y_{2n} - Y_{2n-1}}{c_n} \right) \right) - x^* \right]^2 \\ &\leq \left[ X_n + a_n \left( \frac{Y_{2n} - Y_{2n-1}}{c_n} \right) - x^* \right]^2, \end{aligned}$$

and the inequality follows from the contraction property of Euclidean projection onto any compact interval $I$:

$$(\Pi_I(W_n) - x^*)^2 = (\Pi_I(W_n) - \Pi_I(x^*))^2 \leq (W_n - x^*)^2,$$

with $W_n := X_n + a_n(Y_{2n} - Y_{2n-1})/c_n$. The remainder of the proof is identical.

*for some finite positive constant $C$ which can be explicitly identified. The proof follows almost verbatim the proof in Theorem 1.*

## 2.2 Implications

### 2.2.1 Optimizing the choice of tuning sequences

From Theorem 1, it follows that $c_n \approx a_n^{1/4}$ minimizes the order of the upper bound on the MSE. With this choice Theorem 1 yields an MSE of order $\sqrt{a_n}$. This implies that one should choose $\{a_n\}$ to decrease as "fast" as possible while not violating (S1)-(S4). Proposition 1 shows that $a_n \approx 1/n$ is the optimal choice.

**Proposition 1** *Let the assumptions of Theorem 1 hold and suppose $\{a_n\}$ is a non-increasing sequence. Then the minimal order of the upper bound in (2) is $O(1/\sqrt{n})$, which is achieved by setting $a_n = \theta_a/n$ and $c_n = \theta_c/n^{1/4}$ for any finite positive constants $\theta_a$ and $\theta_c$ with $\theta_a > (\sqrt{2}-1)/(2K_0)$.*

**Remark 7** ***(Optimality of polynomial-like sequences)*** *The result of Proposition 1 recovers the well known optimal rate of convergence of the KW algorithm under assumptions (F1) and (F2); see [9] and [20]. Unlike these papers, as well as essentially all antecedent literature, we do not assume the sequences to have the structure in the proposition, but rather deduce this structure from the more general bounds given in Theorem 1.*

**Remark 8** ***(Specification and adjustment of the tuning sequences)*** *Once the optimal order of tuning sequences has been determined, it is then possible to optimize the constants $\theta_a$ and $\theta_c$. In particular, if we possess a priori knowledge on the curvature of the function $f(\cdot)$ we can specify the sequence $\{a_n\}$ such that the condition $\theta_a > (\sqrt{2}-1)/(2K_0)$ holds, and hence ensure optimal convergence rates for the KW algorithm. Moreover, the explicit expressions for the constants in the upper bounds given in Theorem 1 can be used to further customize $\{a_n\}$ and $\{c_n\}$ so that these constants are optimized. In §3 we show how this idea leads to adaptive modifications of the KW algorithm that are applicable when one does not have good a priori knowledge of the function curvature, Lipschitz bounds, noise level, etc.*

### 2.2.2 Non-necessity of (KW3)

We exhibit sequences $\{a_n\}$ and $\{c_n\}$ which violate assumption (KW3), yet satisfy all assumptions of Theorem 1 and yield convergence of the mean-squared error to zero. This example shows that assumption (KW3) is not necessary to have a convergent KW algorithm under the standard assumptions (F1) and (F2).

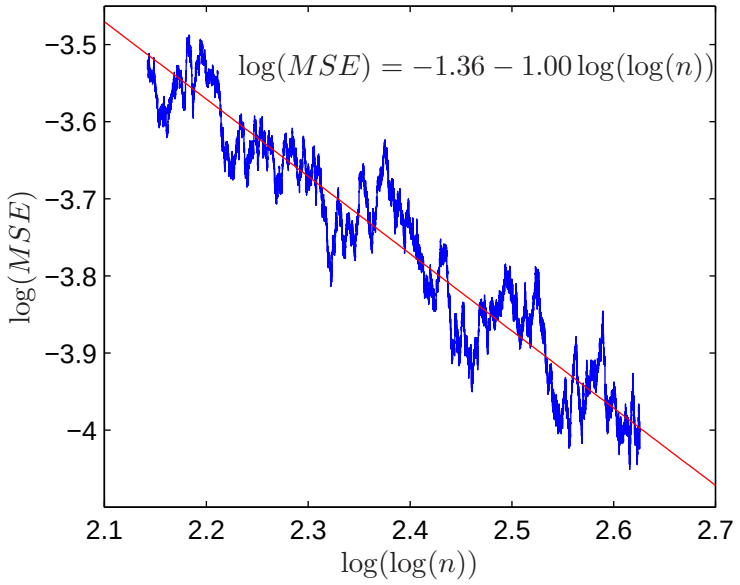


**Figure 1:** Illustration of the non-necessity of (KW3). The figure depicts the behavior of the MSE for a choice of sequences $\{a_n\}$ and $\{c_n\}$ that violates assumption (KW3); the MSE is seen to decay roughly like $(\log(n))^{-1}$, which follows from Theorem 1.

Put $a_n = 1/n$ and $c_n = \sqrt{\log(n+1)/n}$ for $n = 1, 2, \ldots$. It is easily verified that this choice satisfies (S1)-(S4). From Theorem 1 since $c_n^4 < \tau_1 a_n$ with $\tau_1 = 1$, we deduce that the MSE converges to zero at rate $O(a_n/c_n^2) = O(\log(n)^{-1})$ for any function satisfying assumptions (F1) and (F2) and such that $A < 4K_0$. At the same time,

$$\sum_{n=1}^{\infty} \frac{a_n^2}{c_n^2} = \sum_{n=1}^{\infty} \frac{1}{n\log(n+1)} = \infty,$$

which violates (KW3).

Figure 1 gives a plot of log(MSE) versus $\log\log(n)$ for this setting using the function $f(x) = x^2$. To find the MSE at each step, we run the algorithm 1000 times and average the results. The graph shows the results up to $n = 10^6$ steps of the algorithm. For numerical purposes, we assumed additive noise as described in Remark 1 using independent samples of a normal random variable with $\sigma = 1$ at each function evaluation. The regression coefficient in the log-log plot in Figure 1 is for iterations 5000 to $10^6$ and is $-1.0$ (95% confidence interval $(-1.04, -0.97)$), consistent with Theorem 1 which for $a_n = 1/n$ and $c_n = \sqrt{\log(n)/n}$ predicts a convergence rate of $a_n/c_n^2 = 1/\log(n)$.

### 2.3 The special case of "quadratic-like" functions

The paper by Derman [7] analyzes the performance of the KW algorithm for the special case of quadratic-like functions, relying on Chung's lemma and hence restricting $\{a_n\}$ and $\{c_n\}$ to be polynomially decaying sequences. With this restriction the "best" rate of convergence for the MSE is shown to be $O(1/n^{1-\epsilon})$ for some $\epsilon > 0$. Next we revisit this analysis under the general framework developed in §2.

We first restate the assumption given in [7].

(F3) There exist positive constants $K_0, K_1$ and $C_0$ such that for every $c$, with $0 \leq c \leq C_0$,

$$-K_1(x - x^*)^2 \leq \frac{f(x+c) - f(x-c)}{c}(x - x^*) \leq -K_0(x - x^*)^2.$$

**Proposition 2** *Let $\{X_n\}$ be generated by the KW recursion (1) using $\{a_n\}$ and $\{c_n\}$ that satisfy (S1) and (S3) with $A < 2K_0$. Then, under assumption (F3),*

$$\mathbb{E}(X_{n+1} - x^*)^2 \leq Ca_n/c_n^2 \quad \text{for all } n \geq 1, \tag{8}$$

*where $C$ is identified explicitly in (43).*

With this bound in place, we now exhibit the "best" choice of tuning sequences.

**Proposition 3** *Let the assumptions of Proposition 2 hold and suppose $\{a_n/c_n^2\}$ is a non-increasing sequence. Then the minimal order of the upper bound in (8) is $O(1/n)$, which is achieved by setting $a_n = \theta_a/n$ and $c_n = \theta_c$ for any finite positive constants $\theta_a$ and $\theta_c$ satisfying $\theta_a > 1/K_0$.*

**Remark 9** ***(Implications)*** *The result of Proposition 3 indicates that condition (KW1) is not necessary for the class of quadratic-like functions. By not relying on Chung's lemma, we improve on the results of Derman [7] by allowing more general sequences and by eliminating the "for some $\varepsilon > 0$" in his convergence result.*

To illustrate this numerically, let $a_n = 1/n$ and $c_n = 1$. This choice satisfies (S1) with $A = 2$ and (S3) with $\kappa = 1$. By Proposition 2, for any quadratic function, we should observe MSE convergence rate of order $1/n$. In particular for $f(x) = -x^2$, using additive independent standard normal noise at each function evaluation, Figure 2 plots $\log(MSE)$ versus $\log(n)$ up to $n = 10^6$ steps in the algorithm. The regression coefficient for this example is $-0.99$ (95% confidence interval $(-0.997, -1.005)$), which is close to the theoretical value of $-1$ predicted by Proposition 3.

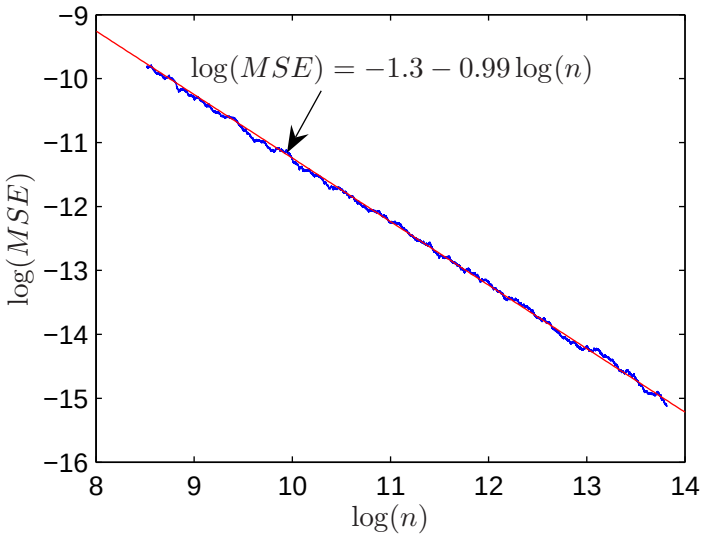


**Figure 2:** Illustration of non-necessity of (KW1) for "quadratic-like" functions. The figure plots $\log(MSE)$ versus $\log(n)$. The MSE behaves roughly like $O(n^{-1})$ which can be calculated using Proposition 2 with $a_n = 1/n$ and $c_n = 1$.

### 2.4 Performance of the KW algorithm under further smoothness assumptions

Dupac [9] derives the optimal rate of convergence for the basic KW algorithm (1) when the underlying function is thrice-differentiable. The result in [9] is restricted to polynomial sequences as it again relies on the Chung's lemma given in [6]. We now revisit this problem and derive an analogue to Theorem 1.

We restrict our attention to functions that satisfy (F1), (F2) and:

(F4) $f'''(x)$ exists for all $x \in \mathbb{R}$ and $|f'''(x)| \leq T$ for some $T \in \mathbb{R}$.

For the sequences to be used in the algorithm we require (S1), (S3) and for some finite positive constants $A, \tau_1$ and $\tau_2$:

(S2') $c_n^4 \leq c_{n+1}^4(1 + Aa_{n+1})$ for all $n \geq 1$.

(S4') Either (i) $c_n^6/a_n \leq \tau_1$, or (ii) $c_n^6/a_n \geq \tau_2$, for all $n \geq 1$.

**Remark 10** *Since the functions are now assumed to be thrice-differentiable, we can expand to one further term in the Taylor expansion and derive a similar recursion to the one used to prove Theorem 1 (see proof sketch there). Hence we require assumptions (S2') and (S4') which replace (S2) and (S4) assumed in §2.*

**Theorem 2** *Let $\{X_n\}$ be generated by the Kiefer-Wolfowitz stochastic approximation recursion given in (1) with $\{a_n\}$ and $\{c_n\}$ satisfying (S1), (S2'), (S3) and (S4') with $A < 4K_0$. Then under*

*assumptions (F1), (F2) and (F4)*

$$\mathbb{E}(X_{n+1} - x^*)^2 \leq \begin{cases} C_1' a_n / c_n^2 & \text{if } c_n^6/a_n \leq \tau_1 \\ C_2' c_n^4 & \text{if } c_n^6/a_n \geq \tau_2, \end{cases} \tag{9}$$

*for all $n \geq 1$ and for some finite positive constants $C_1'$ and $C_2'$.*

The proof follows the same steps as in the proof of Theorem 1. The main difference is in the first step where we derive bounds on the gradient estimate using further smoothness assumed here. This adds one more term in the Taylor expansion of step 1 in the proof outline of Theorem 1, and in turn modifies the real number recursion for $b_n$ outlined there.

Theorem 2 suggests that one should set $c_n \approx a_n^{1/6}$ to minimize the upper bound, whose order is then $O(a_n^{2/3})$. This implies that one should choose $\{a_n\}$ to decrease as "fast" as possible while not violating (S1), (S2'), (S3) and (S4') to get the optimal rate. The best choice of the tuning sequences is given as follows.

**Proposition 4** *Let the assumption of Theorem 2 hold and suppose $\{a_n\}$ is a non-increasing sequence. Then the minimal order of the upper bound in (9) is $O(n^{-2/3})$, which is achieved by the setting $a_n = \theta_a/n$ and $c_n = \theta_c/n^{1/6}$ for any finite positive constants $\theta_a$ and $\theta_c$ that satisfy $\theta_a > (2^{2/3} - 1)/(2K_0)$.*

# 3 Finite-Time Behavior

## 3.1 Problems and remedies for finite-time behavior

Despite theoretical performance guarantees (e.g., those contained in Theorem 1), it is well known that stochastic approximation methods often perform quite poorly in practice. This emphasizes the importance of investigating the *finite-time behavior* of the algorithm, to complement the long run asymptotics and rates of convergence.

In this section we propose a modified version of the KW algorithm, which we call the *Scaled-and-Shifted KW algorithm*. This algorithm uses simple adaptive adjustments of the tuning sequences to address three main sources of poor performance:

1. a long oscillatory period due to a step-size sequence $\{a_n\}$ that is "too large;"
2. a degraded convergence rate due to a step-size sequence $\{a_n\}$ that is "too small;"
3. poor gradient estimates due to a gradient estimation step-size sequence $\{c_n\}$ that is "too small."

Next we explain in more detail each of these problems, illustrate them numerically and propose potential remedies that are combined in the final scaled-and-shifted KW algorithm.

### 3.1.1 The oscillation problem

An issue that can arise in practical applications of the truncated KW algorithm (which is described in Remark 4) is a long period characterized by oscillations between boundaries of the truncation interval.

**Definition 1** ***(Oscillatory period)*** *Consider the truncated KW algorithm restricted to an interval $I_0 = [l, u]$. The oscillatory period $T$ is defined as the number of iterations until the algorithm ceases consecutive visits to different boundary points, i.e.,*

$$T = \sup\{n \geq 2 : (X_n = u - c_n \text{ and } X_{n-1} = l + c_{n-1}) \text{ or } (X_n = l + c_n \text{ and } X_{n-1} = u - c_{n-1})\}, \quad (10)$$

*if the supremum on the right-hand-side above is finite, otherwise we set $T = 0$.*

Roughly speaking, when the step-size sequence $\{a_n\}$ is too large relative to the gradient, the algorithm will exhibit a long transient period oscillating between boundary points until the step size becomes suitably small. This issue will not affect the algorithm's asymptotic performance, but the following example illustrates the severity of the problem. Figure 3(a) shows a single path of the truncated KW algorithm using $I_0 = [-50, 50]$ for the function $f(x) = -x^4$, and independent standard normal additive noise (i.e., $Y_i = f(x) + \varepsilon_i$, with $\varepsilon_i \sim N(0, \sigma^2)$ and $\sigma = 1$) and $X_1 = 30$. The tuning sequences are $a_n = 1/n$ and $c_n = 1/\sqrt[4]{n}$ as prescribed in §2. The oscillatory behavior can be observed for the first $T = 9960$ iterations and the algorithm only starts to converge after this period. The relative frequency of $X_{10,000}$ over many paths is illustrated in Figure 3(b). Even after 10,000 iterations, most of the paths are relatively far from $x^* = 0$.

The length $T = 9960$ of the oscillatory period depends on the length of the initial interval $I_0$. If one has more a priori information about the point of maxima and can specify a smaller initial interval, then the oscillatory period will be shorter. Similarly, less a priori information requires a larger initial interval, which leads to a longer oscillatory period. Figure 4 exhibits the relation between the average length of the oscillatory period estimated over 1000 sample paths and the length of the initial interval for the function $f(x) = -x^4$.

The long oscillatory period is caused by a step-size sequence $\{a_n\}$ that is too large in comparison to the magnitude of the gradient. To avoid this, we propose to decrease the step size when necessary by shifting the $\{a_n\}$ sequence; i.e., redefining the sequence $\{a'_n\} := \{a_{n+\beta}\}$ for some positive integer $\beta$. Specifically, whenever an iterate $X_n$ falls outside the truncation interval known to contain $x^*$, we

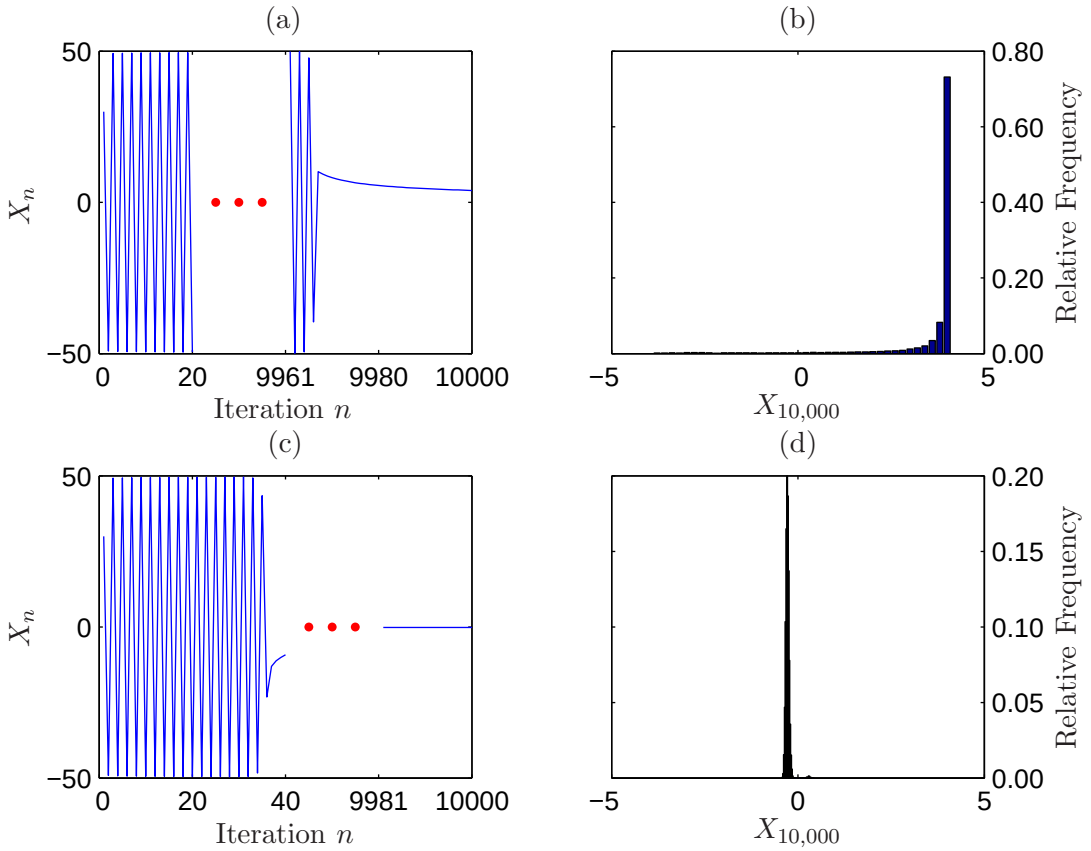


**Figure 3:** Oscillatory behavior of the truncated KW algorithm. Panel (a) shows a sample path of iterates in the truncated KW algorithm for the function $f(x) = -x^4$ with $a_n = 1/n$, $c_n = 1/\sqrt[4]{n}$ and $\sigma = 1$. The initial interval is assumed to be $I_0 = [-50, 50]$ and oscillatory behavior is observed for $T = 9960$ iterations. Panel (c) shows a sample path in the scaled-and-shifted KW algorithm in the same setting, using the same noise random sequence. The shift of $\beta = 9800$ corresponding to $a_n = 1/(n + 9800)$ is finalized after 28 iterations. The sequence $c_n = 1/\sqrt[4]{n}$ is not shifted. Panels (b) and (d) give the relative frequency of $X_{10,000}$ using 15,000 simulation replications for the truncated KW and scaled-and-shifted KW algorithms, respectively.

calculate the minimum positive integer $\beta$ so that using $a_{n+\beta}$ ensures that the function evaluations are within the interval; i.e., both $X_n \pm c_n \in [l, u]$. The shifted sequence is used in the computation of all future iterates. Multiple shifts can occur, but the number of shifts is bounded in advance. Note that the shift(s) is adaptive, i.e., it is determined during the course of the algorithm and it does not require any additional information about the function. Figure 3(c) presents a typical sample path that results from applying the shift using the same parameters and random numbers as in Figure 3(a) and Figure 3(d) gives the relative frequency chart for $X_{10,000}$ using 15,000 simulation replications.

**Remark 11** ***(Intuition for shifting)*** *The idea of shifting the* $\{a_n\}$ *sequence is inspired by close examination of the constants in the upper bounds developed in* §*2. For instance, if we try to minimize*

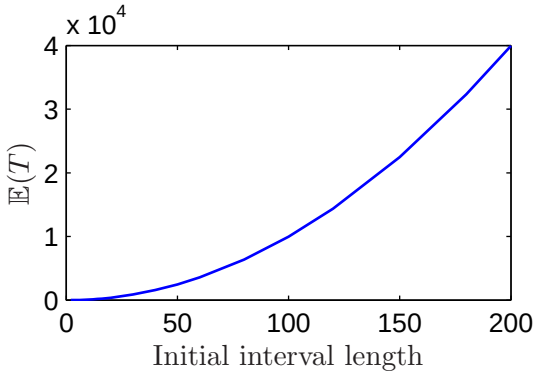


**Figure 4:** Oscillatory period vs. initial interval. This figure shows the estimated average length of the oscillatory period $T$ as a function of the initial interval length $u - l$, for the function $f(x) = -x^4$ with $a_n = 1/n$, $c_n = 1/\sqrt[4]{n}$ and $\sigma = 1$.

*the constant in the upper bound for "quadratic-like" functions (43), we observe that a balancing between the two terms is required. Using (40), the first term in (43) decreases with a decrease in the $\{a_n\}$ sequence for large values of $K_1$, which corresponds to a steep gradient. On the other hand, the second term increases as $\{a_n\}$ decreases, so there is a tradeoff. The key observation is that for large values of $K_1$, the first term dominates the second one and therefore having a smaller $\{a_n\}$ sequence decreases the value of $C$. Decreasing the step-size sequence $\{a_n\}$ by a shift rather than a multiplicative factor preserves more "energy" for future iterations.*

### 3.1.2 Degraded convergence rate due to a small step size

The asymptotic results developed in the literature, as well as the bounds given in Theorem 1, require a careful choice of the $\{a_n\}$ sequence in relation to the curvature of the function that is being optimized. This is encoded in assumptions (S1) and (S2) with the requirement that $A < 4K_0$; see also [11] for further discussion. If the tuning sequences do not satisfy this assumption, for instance if the multiplicative constant $\theta_a$ in $a_n = \theta_a/n$ is not large enough, a degraded convergence rate may result. As a simple example, similar to the one worked out in [11], consider $f(x) = -0.001x^2$ with $a_n = 1/n$ and $c_n = 1/\sqrt[4]{n}$, and there is no observation error (i.e., $\sigma = 0$). Then the KW recursion becomes $X_{n+1} = X_n(1 - 1/(250n))$. Starting with $X_1 = 30$, we have

$$X_n = 30 \prod_{m=1}^{n-1} \left(1 - \frac{1}{250m}\right) \geq \exp\left[-\sum_{m=1}^{n-1} \frac{1}{250m - 1}\right] \geq \frac{27}{n^{0.004}}, \tag{11}$$

so the MSE cannot converge faster than $27^2/n^{0.008}$. In contrast, the upper bound in Theorem 1 guarantees a rate of $1/\sqrt{n}$, but this rate is not achieved because the $\{a_n\}$ sequence violates (S1) and (S2). Figure 5(a) illustrates a sample path of the iterates $X_n$ in this setup with independent normal noise with zero mean and standard deviation $\sigma = 0.001$. The MSE convergence rate for this setting

is $-0.008$ (see Table 3 for a corresponding confidence interval) which matches the theoretical rate given in (11). The relative frequency of $X_{10,000}$ given in Figure 5(b) shows all sample paths exhibit a similar lack of convergence.

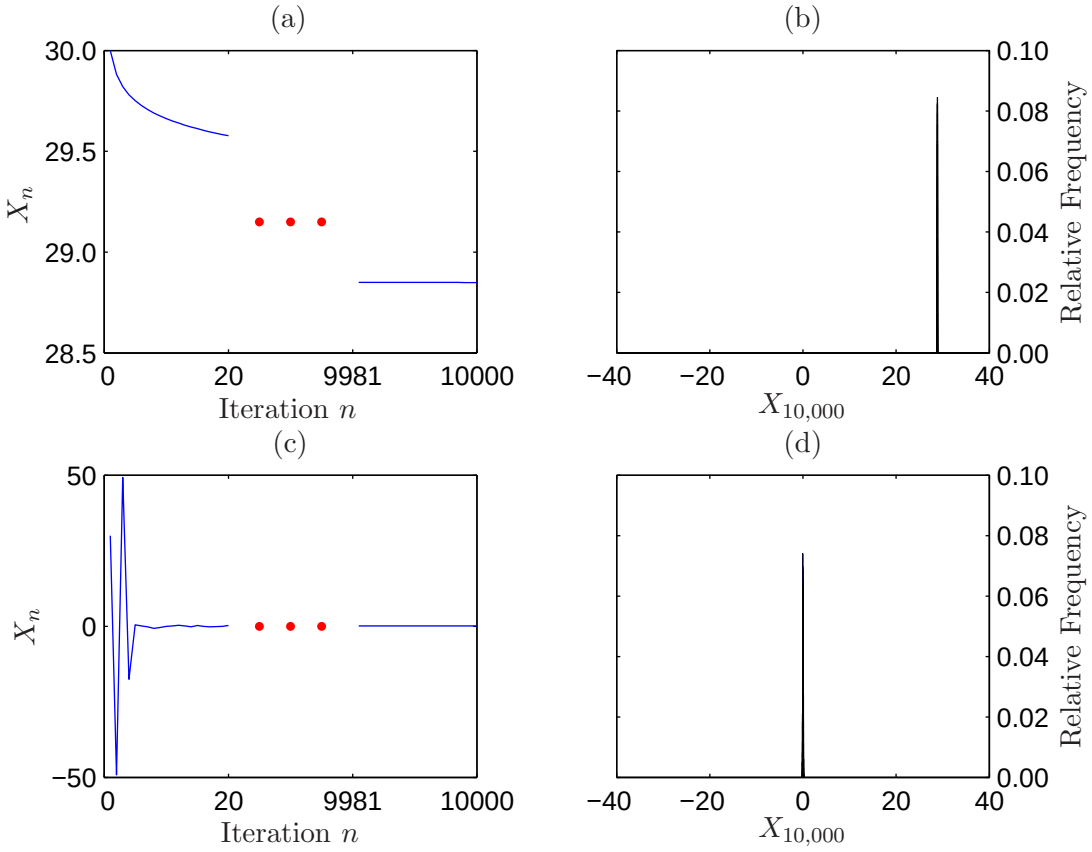


**Figure 5:** Degraded convergence rate due to a small step size. Panel (a) shows a sample path in the KW algorithm for $f(x) = -0.001x^2$ with $a_n = 1/n$, $c_n = 1/\sqrt[4]{n}$ and $\sigma = 0.001$. The $A < 4K_0$ assumption in (S1) and (S2) is violated. From Table 3, the convergence rate of the MSE is $-0.008 \pm 3.2 \times 10^{-8}$. Panel (b) is the relative frequency chart for $X_{10,000}$ and it shows the slow convergence behavior in all 15,000 simulated sample paths. Panel (c) shows a sample path in the scaled-and-shifted KW algorithm under the same setting using the same noise random sequence. After two scale-ups, the $\{a_n\}$ sequence becomes $a_n = 1002/n$ and shifting is not needed. From Table 3, the scaling results in an MSE convergence rate of $-0.501 \pm 0.007$ which recovers the optimal rate of $-0.5$. As seen in panel (d), this recovery can be observed in all of the 15,000 simulated sample paths.

Our remedy for this rate degradation problem is to scale up the $\{a_n\}$ sequence as follows. In the first several iterations of the algorithm, we multiply the $\{a_n\}$ sequence by a constant greater than or equal to one, so that iterate $n$ is at the boundary of the current truncation interval, i.e., $X_n = l + c_n$ or $X_n = u - c_n$. This scaling up forces the algorithm to oscillate between the endpoints of the truncation interval $I_n = [l + c_n, u - c_n]$. This maps the problem of rate degradation into a problem of oscillatory behavior, which is then remedied by the shifted sequence approach of §3.1.1. The number of potential scale-ups is limited to a small finite number (two in our numerical examples).

Figure 5(c) shows a sample path of iterates generated by the scaled-and-shifted KW algorithm on $f(x) = -0.001x^2$ using the same parameters and same random numbers as in Figure 5(a). In this example, no shifting is needed after the $\{a_n\}$ sequence is scaled up and the optimal rate of convergence is recovered with this simple scaling (see Table 3 for a confidence interval on the convergence rate). As seen in Figure 5(d), the scaled-and-shifted KW algorithm improves the convergence on all 15,000 simulated samples.

### 3.1.3 The problem of noisy gradient estimates

The finite difference estimate of the gradient in (1) uses a tuning sequence $\{c_n\}$. Cases where the noise in the function observation is too large in magnitude relative to the $\{c_n\}$ sequence may give rise to excessive noise in the gradient estimates. As a consequence, even at the boundaries of the truncation interval, the algorithm may step away from the point of maximum of the function. Moreover, the iterates might move in random directions governed purely by the noise for a long period of iterations. This can lead to poor finite-time performance, even if the asymptotic convergence rate is eventually achieved. Figure 6(a) illustrates a sample path for the function $f(x) = 1000\cos(\pi x/100)$ with $a_n = 1/n$, $c_n = 1/n^{1/4}$ and an initial interval $I_0 = [-50, 50]$. As before, we assume independent normal additive noise, i.e., $Y_i = f(x) + \varepsilon_i$, with $\varepsilon_i \sim N(0, \sigma^2)$ and $X_1 = 30$. The main difference is that we assume a large noise level given by $\sigma = 1000$. The sample path in Figure 6(a) does not show convergent behavior for the first 10,000 iterations. (Similar behavior can be observed even up to 100,000 iterations.) The relative frequency of $X_{10,000}$ in Figure 6(b) shows a nearly uniform distribution between $-50$ and 50, i.e., the algorithm has not improved over $X_1$ in 10,000 iterations.

Our remedy for this problem is to scale up the $\{c_n\}$ sequence. Specifically, we multiply the $\{c_n\}$ sequence by a constant $\gamma_0 > 1$ when an iterate hits the boundary of the interval and the gradient estimate points in a direction away from the current truncation interval (i.e., away from $x^*$). We also make sure that the scaled-up $\{c_n\}$ sequence does not exceed an upper bound $c_{max}$, which is a parameter for our algorithm. In our numerical examples, we use $\gamma_0 = 2$. Multiple scale-ups can occur, but the number is bounded in advance. The scaled-up $\{c_n\}$ sequence is used for the remaining iterations of the algorithm. The $\{a_n\}$ sequence is also scaled and shifted as necessary as described before. Figure 6(c) shows the sample path of the scaled-and-shifted KW algorithm applied to the function $f(x) = 1000\cos(\pi x/100)$ with the same parameters and random numbers. The $\{c_n\}$ sequence is scaled up four times at early stages of the algorithm, while the $\{a_n\}$ sequence is shifted but not scaled. With this adaptive tuning of the sequences, the iterates move toward the point of maximum much faster and this behavior is consistent throughout 15,000 sample paths as shown in Figure 6(d). In this setting, the scaled-and-shifted KW algorithm achieves an MSE convergence rate of $-0.490 \pm 0.004$ (see Table 5).

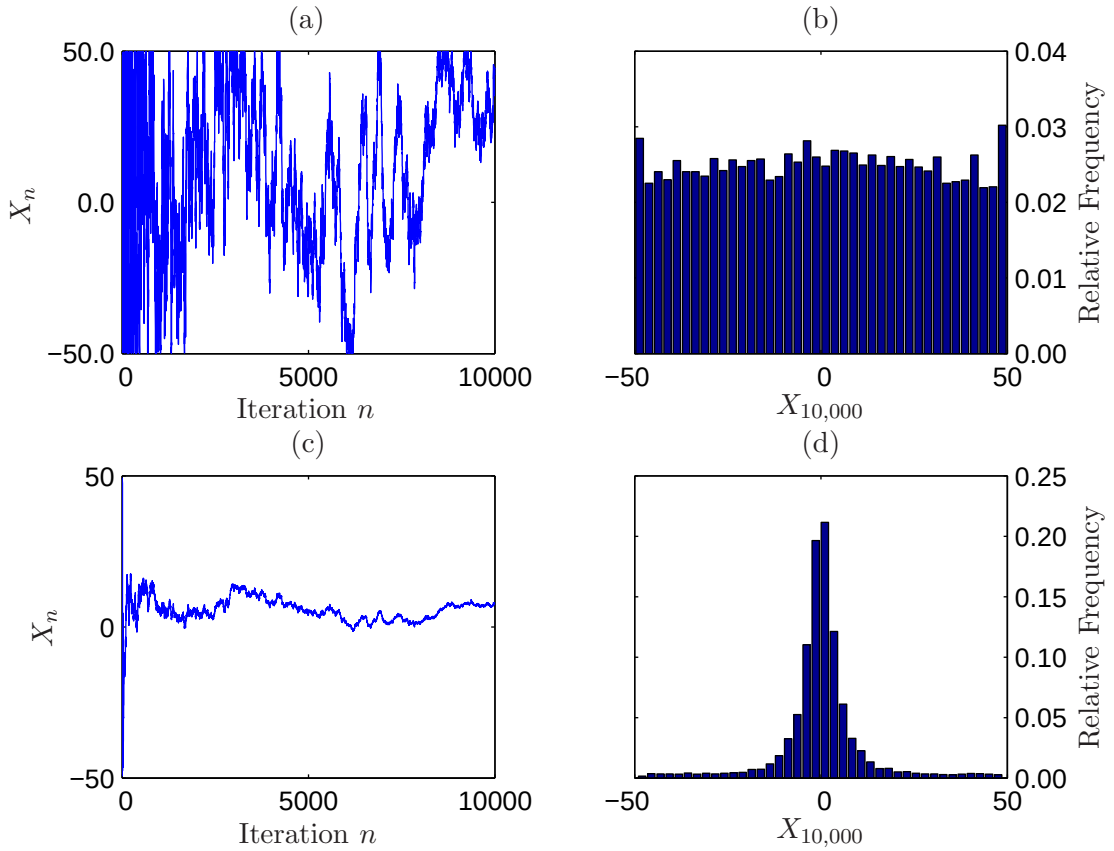


**Figure 6:** Noisy gradient estimate problem. Panel (a) shows a sample path of the truncated KW algorithm for the function $f(x) = 1000\cos(\pi x/100)$ assuming an initial interval $I_0 = [-50, 50]$, using standard normal noise with $\sigma = 1000$, $a_n = 1/n$ and $c_n = 1/n^{1/4}$. The MSE convergence rate for this setting is $-0.051 \pm 0.002$ (see Table 5). Panel (c) shows a sample path in the scaled-and-shifted KW algorithm for the same function, using the same noise random sequence. The algorithm adjusts both tuning sequences to the noise level and shows much faster convergence. After scaling and shifting, the final sequences are $a_n = 1/(n + 61)$ and $c_n = 16/n^{1/4}$. The last row in Table 5 corresponds to this setting and gives an MSE rate of convergence of $-0.490 \pm 0.004$. The relative frequency of $X_{10,000}$ in both algorithms are given in panels (b) and (d).

### 3.2 Scaled-and-shifted KW algorithm

We present a formal statement of the scaled-and-shifted KW algorithm. As in the truncated KW algorithm, we assume knowledge of an interval $I_0 = [l, u]$, that contains the point of maximum. The scaled-and-shifted KW algorithm requires parameter inputs to be set by the user in Step 1.

**Step 1. Set algorithm parameters**

- $h_0$: the number of forced hits to boundary points $l + c_n$ and $u - c_n$ by scaling up the $\{a_n\}$ sequence (sample default: 2).
- $\gamma_0$: the scaling up factor for the $\{c_n\}$ sequence (sample default: 2).

- $k_a$: an upper bound on the number of shifts in the $\{a_n\}$ sequence (sample default: 30).
- $\upsilon_a$: used to upper bound the amount of the shift in the $\{a_n\}$ sequence (sample default: $(u-l)/10{,}000$).
- $k_c$: an upper bound on the number of scale-ups in the $\{c_n\}$ sequence (sample default: 20).
- $c_0$: the parameter defining the maximum possible value of $\{c_n\}$ sequence after the scale-ups; i.e., $c_n \leq c_{max} = c_0(u-l)$ for all $n \geq 1$ (sample default: 0.2).
- $m_{max}$: an upper bound on the iteration number of the last adaptation in the sequences, i.e., after iteration $m_{max}$ no scaling nor shifting on the sequences is done; we require $m_{max} \geq h_0$ (sample default: total number of iterations).
- $X_1$: initial starting point; can be random or deterministic but must be in the interval $[l + c_1, u - c_1]$.

**Step 2. Initialization**

Set

- $s_a = 0$: a variable keeping track of the number of shifts in the $\{a_n\}$ sequence.
- $s_c = 0$: a variable keeping track of the number of scale-ups in the $\{c_n\}$ sequence.

**Step 3:** For $n \leq h_0$,

(a) Calculate $X_{n+1}$ using the KW recursion given in (1).

(b) Scale up the $\{a_n\}$ sequence, if necessary, ensuring a different interval endpoint is hit:

(i) If $X_{n+1} < u - c_{n+1}$ and $X_{n+1} > X_n$, find the scale-up factor for $\{a_n\}$ sequence that makes $X_{n+1} = u - c_{n+1}$; i.e., set $\alpha = (u - c_{n+1} - X_n)/(X_{n+1} - X_n)$ and then use the new sequence $\{a_n\} \leftarrow \{\alpha a_n\}$ for the rest of the iterations.

(ii) If $X_{n+1} > l + c_{n+1}$ and $X_{n+1} < X_n$, find the scale-up factor for $\{a_n\}$ sequence that makes $X_{n+1} = l + c_{n+1}$; i.e., set $\alpha = (l + c_{n+1} - X_n)/(X_{n+1} - X_n)$ and then use the new sequence $\{a_n\} \leftarrow \{\alpha a_n\}$ for the rest of the iterations.

(c) Scale up the $\{c_n\}$ sequence whenever the gradient estimate is too noisy:

(i) If $X_{n+1} > u - c_{n+1}$, $X_n = u - c_n$ and $s_c \leq k_c$ then set $s_c \leftarrow s_c + 1$, calculate $\gamma = \min\{\gamma_0, c_{max}/c_{n+1}\}$ and use the new sequence $\{c_n\} \leftarrow \{\gamma c_n\}$ for the rest of the iterations.

(ii) If $X_{n+1} < l + c_{n+1}$, $X_n = l + c_n$ and $s_c \leq k_c$ then set $s_c \leftarrow s_c + 1$, calculate $\gamma = \min\{\gamma_0, c_{max}/c_{n+1}\}$ and use the new sequence $\{c_n\} \leftarrow \{\gamma c_n\}$ for the rest of the iterations.

(d) Set $X_{n+1} = \min\{u - c_{n+1}, \max\{X_{n+1}, l + c_{n+1}\}\}$. Increment $n$.

**Step 4:** For $n > h_0$,

(a) Calculate $X_{n+1}$ using the KW recursion given in (1).

(b) If $n \leq m_{max}$, shift the $\{a_n\}$ sequence, if necessary, to prevent iterates exiting the truncation interval, but use the upper bound parameter $\upsilon_a$ to prevent the shift from being too large.

(i) If $X_{n+1} > u - c_{n+1}$, $X_n < u - c_n$ and $s_a \leq k_a$ then set $s_a \leftarrow s_a + 1$ and find the shift in $\{a_n\}$ sequence that makes $X_{n+1} \leq u - c_{n+1}$; i.e.,

- Solve $\max(u - c_{n+1} - X_n, \upsilon_a) / \left(\frac{Y_{2n} - Y_{2n-1}}{c_n}\right) = a_{n+\beta}$ for $\beta$
- Use $\{a_n\} \leftarrow \{a_{n+\lceil\beta\rceil}\}$ for the rest of the iterations

(ii) If $X_{n+1} < l + c_{n+1}$, $X_n > l + c_n$ and $s_a \leq k_a$ then set $s_a \leftarrow s_a + 1$ and find the shift in $\{a_n\}$ sequence that makes $X_{n+1} \leq l + c_{n+1}$; i.e.,

- Solve $\min(l + c_{n+1} - X_n, -\upsilon_a) / \left(\frac{Y_{2n} - Y_{2n-1}}{c_n}\right) = a_{n+\beta}$ for $\beta$
- Use $\{a_n\} \leftarrow \{a_{n+\lceil\beta\rceil}\}$ for the rest of the iterations

(c) If $n \leq m_{max}$, repeat Step 3(c).

(d) Set $X_{n+1} = \min\{u - c_{n+1}, \max\{X_{n+1}, l + c_{n+1}\}\}$. Increment $n$.

One should note that when we scale or shift one of the two tuning sequences, the modification is made only on that sequence. For instance, if the $\{a_n\}$ sequence is shifted to $\{a_{n+\beta}\}$ at any iteration, we still continue using $c_n$, not $c_{n+\beta}$.

The result given in Theorem 1 also hold for the scaled-and-shifted KW algorithm. As mentioned in Remark 2, it is enough to have conditions (S1)-(S4) satisfied "for all sufficiently large $n$" for the bounds in Theorem 1 to hold. Since neither scaling nor shifting can occur after iteration $m_{max}$, the conditions of Theorem 1 hold for all $n > m_{max}$. Together with the extension of Theorem 1 to the truncated KW algorithm (see Remark 4), this is enough to conclude that adaptation of the tuning sequences in the scaled-and-shifted KW algorithm preserve the theoretical asymptotic guarantees.

**Remark 12** ***(Shifting in the special case of*** $a_n = 1/n$***)*** *We have described how to calculate the shift in the* $\{a_n\}$ *sequence in Step 4(c) for arbitrary* $\{a_n\}$*. Now suppose* $a_n = 1/n$ *is chosen*

*as the sequence and suppose it became* $a_n = \theta_a/(n+s)$ *for some* $\theta_a \in \mathbb{R}^+$ *and* $s \in \mathbb{Z}^+$ *after some scaling and shifting in the scaled-and-shifted KW algorithm. Then* $\beta$ *is given by:*

$$\beta = \frac{\theta_a}{a_n}\left[\frac{X_{n+1} - X_n}{\max(\upsilon_a, u - c_{n+1} - X_n)} - 1\right]$$

*in Step 4(i) and*

$$\beta = \frac{\theta_a}{a_n}\left[\frac{X_{n+1} - X_n}{\min(-\upsilon_a, l + c_{n+1} - X_n)} - 1\right]$$

*in Step 4(ii).*

### 3.3 Numerical results

In this section we provide numerical results for the scaled-and-shifted KW algorithm as described in §3.2 which combines the remedies described previously. The MATLAB code and documentation can be downloaded from *www.columbia.edu/∼mnb2/broadie/research.html*. Results for the truncated KW algorithm are given for comparison. Algorithm sample paths were generated for 10,000 iterations. MSE results are computed with 15,000 independent replications of the algorithm and the standard errors are within 0.5% of the MSE values in all cases. Empirical convergence rates were calculated by computing a least squares fit of $\log(MSE)$ vs. $\log(n)$ using iterations $n$ = 5,000 to $n$ = 10,000. Final performance measures and confidence intervals were computed from 50 independent tests.

The initial tuning sequences were $a_n = 1/n$ and $c_n = 1/\sqrt[4]{n}$ in all cases. We report the statistics on the final adapted tuning sequences in separate tables. The initial interval used in all examples is $[-50, 50]$ and the initial starting point is always set to $X_1 = 30$. The input parameters for the scaled-and-shifted KW algorithm are set so that we hit the boundaries of the truncation interval at the first two iterations ($h_0 = 2$), and the scale-up factor for the $\{c_n\}$ sequence is two ($\gamma_0 = 2$). The upper bounds on the total number of shifts in the $\{a_n\}$ sequence and scale-ups in the $\{c_n\}$ sequence are set to be $k_a = 50$ and $k_c = 50$ respectively. In order to upper bound the amount of shift in single iteration, we set $\upsilon_a = (u - l)/10{,}000$. Also $c_0 = 0.2$ is used so that $c_n \leq 20$ holds for all $n \geq 1$. All functions are estimated using $Y_i = f(x) + \varepsilon_i$ with independent noise $\varepsilon_i \sim N(0, \sigma^2)$.

**Table 1:** Comparison of the scaled-and-shifted KW algorithm and the truncated KW algorithm for $f(x) = -x^4$. This table shows the MSE calculated at 50, 500 and 5000 iterations, the convergence rate for the MSE and the 5th and 95th percentile along with the median of the length of the oscillatory periods at different noise levels ($\sigma$). The numbers in square brackets [·] correspond to the truncated KW algorithm.

| | MSE | | | Length of oscillatory period | | |
|---|---|---|---|---|---|---|
| $\sigma$ | 50 | 500 | 5000 | 5% | Median | 95% |
| 0.1 | 30.98 | 1.30 | 0.14 | 26 | 26 | 26 |
| | [2463] | [2479] | [2488] | [9960] | [9960] | [9960] |
| 1 | 30.23 | 1.30 | 0.14 | 26 | 26 | 28 |
| | [2463] | [2479] | [2488] | [9959] | [9960] | [9960] |
| 10 | 22.18 | 1.20 | 0.18 | 22 | 26 | 30 |
| | [2463] | [2479] | [2488] | [9957] | [9959] | [9961] |

**Table 2:** Tuning sequence statistics for $f(x) = -x^4$. The 5th and 95th percentile along with the median values are given for the scale-up factor $\alpha$, and the shift amount $\beta$, in the $\{a_n\}$ tuning sequence, as well as the scale-up factor $\gamma$, for the $\{c_n\}$ sequence. For this test function, we observe only shifting of the $\{a_n\}$ sequence and no scaling up at all noise levels ($\sigma$).

| | $\alpha$ | | | $\beta$ | | | $\gamma$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $\sigma$ | 5% | Median | 95% | 5% | Median | 95% | 5% | Median | 95% |
| 0.1 | 1 | 1 | 1 | 9799 | 9799 | 9799 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 9799 | 9799 | 9800 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 9796 | 9799 | 9801 | 1 | 1 | 1 |

**Table 3:** Comparison of the scaled-and-shifted KW algorithm and the truncated KW algorithm for $f(x) = -0.001x^2$. This table shows the MSE calculated at 50, 500 and 5000 iterations, the convergence rate for the MSE, and the 5th and 95th percentile and the median of the length of the oscillatory periods at different noise levels, $\sigma$. The numbers in square brackets [·] correspond to the truncated KW algorithm.

| | MSE | | | Convergence | Length of oscillatory period | | |
|---|---|---|---|---|---|---|---|
| $\sigma$ | 50 | 500 | 5000 | Rate | 5% | Median | 95% |
| 0.001 | 0.039 | 0.012 | 0.004 | $-0.501 \pm 0.007$ | 2 | 2 | 2 |
| | [868] | [852] | [837] | $[-0.008 \pm 3.2 \times 10^{-8}]$ | [0] | [0] | [0] |
| 0.01 | 4.0 | 1.2 | 0.4 | $-0.501 \pm 0.007$ | 2 | 2 | 2 |
| | [868] | [852] | [837] | $[-0.008 \pm 3.2 \times 10^{-7}]$ | [0] | [0] | [0] |
| 0.1 | 280 | 94 | 31 | $-0.479 \pm 0.007$ | 2 | 2 | 4.7 |
| | [868] | [852] | [837] | $[-0.008 \pm 3.2 \times 10^{-6}]$ | [0] | [0] | [0] |
| 1 | 753 | 393 | 158 | $-0.470 \pm 0.004$ | 2 | 2 | 4.8 |
| | [873] | [857] | [842] | $[-0.008 \pm 3.2 \times 10^{-5}]$ | [0] | [0] | [0] |

**Example 1.** The first test function is $f(x) = -x^4$. This function does not satisfy assumption (F1) and hence we do not have a theoretical MSE convergence rate, but it serves to "stress test" the algorithm. When the truncated KW algorithm is applied to this function, slow convergence is often observed due to long oscillatory periods. Table 1 shows this effect and also shows that the scaled-and-shifted KW algorithm decreases the oscillatory period significantly for all noise levels

and dramatically reduces the MSE. Statistics on the adaptations to the sequences which achieve these results are given in Table 2.

**Example 2.** The second test function is $f(x) = -0.001x^2$, which has a "flat" gradient away from the point of maximum. The $\{a_n\}$ sequence then violates assumption $A < 4K_0$ of Theorem 1. This results in a degraded rate of convergence which also impacts the finite-time behavior of the algorithm. Table 3 shows that the scaled-and-shifted KW algorithm significantly improves the convergence rate, while the observed convergence rate of truncated KW algorithm is close to zero, i.e., it does not converge in practice. Table 4 presents statistics on the adaptations to the sequences which achieve these results. There is significant scaling up in the $\{a_n\}$ sequence, especially for small noise levels. The scaling up in $\{c_n\}$ sequence is pronounced at high noise levels. Although there is no shifting in $\{a_n\}$ sequence at small $\sigma$ values, when $\sigma$ gets larger, we observe occasional large shifts as shown in the 95% values for $\beta$.

**Table 4:** Tuning sequence statistics for $f(x) = -0.001x^2$. The 5th and 95th percentile along with the median values are given for the scale-up factor $\alpha$, and the shift amount $\beta$, in the $\{a_n\}$ sequence, as well as the scale-up factor $\gamma$, for the $\{c_n\}$ sequence, for various noise levels ($\sigma$).

| | $\alpha$ | | | $\beta$ | | | $\gamma$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $\sigma$ | 5% | Median | 95% | 5% | Median | 95% | 5% | Median | 95% |
| 0.001 | 987 | 1001 | 1015 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0.01 | 878 | 1001 | 1165 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0.1 | 476 | 1119 | 9170 | 0 | 4 | 964 | 1 | 2 | 4 |
| 1 | 75 | 298 | 3249 | 0 | 33 | 9811 | 2 | 8 | 32 |

**Table 5:** Comparison of the scaled-and-shifted KW algorithm and the truncated KW algorithm for $f(x) = 1000\cos(\pi x/100)$. This table shows the MSE calculated at 50, 500 and 5000 iterations, the convergence rate for the MSE, and the 5th and 95th percentile and the median of the length of the oscillatory periods at different noise levels ($\sigma$). The numbers in square brackets $[\cdot]$ correspond to the truncated KW algorithm.

| | MSE | | | Convergence | Length of oscillatory period | | |
|---|---|---|---|---|---|---|---|
| $\sigma$ | 50 | 500 | 5000 | Rate | 5% | Median | 95% |
| 10 | 28.5 | 8.3 | 2.6 | $-0.502 \pm 0.006$ | 2 | 2 | 3 |
| | [8.5] | [2.6] | [0.8] | $[-0.505 \pm 0.008]$ | [0] | [0] | [0] |
| 100 | 408 | 142 | 42 | $-0.580 \pm 0.009$ | 2 | 2 | 5 |
| | [645] | [287] | [87] | $[-0.532 \pm 0.008]$ | [0] | [0] | [3] |
| 1000 | 813 | 456 | 187 | $-0.490 \pm 0.004$ | 2 | 3 | 5 |
| | [1744] | [1047] | [840] | $[-0.051 \pm 0.002]$ | [30] | [62] | [116] |

**Example 3.** The last test function is $f(x) = 1000\cos(\pi x/100)$; this specification enables us to use the same truncation interval $[-50, 50]$ used in the two other cases. Note that the function satisfies conditions (F1) and (F2) in the truncation interval. Table 5 shows that the scaled-and-shifted KW algorithm outperforms the truncated KW algorithm in both MSE and convergence rate measures for large noise levels. The only case where the truncated KW algorithm outperforms its adaptive

counterpart in terms of MSE is at the lower noise level of $\sigma = 10$. In this case, since the assumption $A < 4K_0$ is satisfied for the initial choice of the $\{a_n\}$ sequence, the scaling up of the $\{a_n\}$ sequence decreases performance in terms of MSE. But since the algorithm does not "know" the assumption holds, it forces the iterates to hit the boundary at the first two iterations by increasing the step-size sequence $\{a_n\}$. Although the rate of convergence is still preserved, we observe slightly worse MSE results. Statistics about the adaptation of the sequences are given in Table 6.

**Table 6:** Modifications in the tuning sequences for $f(x) = 100\cos(\pi x/100)$. The 5th and 95th percentile along with the median values are given for the scale-up factor, $\alpha$ and the shift amount, $\beta$ in $\{a_n\}$ sequence as well as the scale-up factor for the $\{c_n\}$ sequence, $\gamma$, for various noise levels ($\sigma$).

| | $\alpha$ | | | $\beta$ | | | $\gamma$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $\sigma$ | 5% | Median | 95% | 5% | Median | 95% | 5% | Median | 95% |
| 10 | 2.2 | 3.2 | 5.7 | 0 | 0 | 2 | 1 | 1 | 1 |
| 100 | 1.0 | 2.1 | 20.0 | 0 | 13 | 3113 | 1 | 4 | 8 |
| 1000 | 1.0 | 1.0 | 4.6 | 1 | 167 | 38220 | 8 | 16 | 52 |

# A Proofs

## A.1 Proof of Theorem 1

**Step 1:** Fix $\{a_n\}$ and $\{c_n\}$ as in the statement of the theorem. For positive integer $n$ and $x_n \in \mathbb{R}$, using Taylor expansion, there exist $0 \leq T_1, T_2 \leq 1$ such that

$$\begin{aligned} f(x_n + c_n) &= f(x_n) + f'(x_n + T_1 c_n) c_n \\ f(x_n - c_n) &= f(x_n) - f'(x_n - T_2 c_n) c_n. \end{aligned}$$

Using this, we have

$$\begin{aligned} \widehat{\nabla} f(x_n) &:= \frac{f(x_n + c_n) - f(x_n - c_n)}{c_n} \qquad (12) \\ &= f'(x_n + T_1 c_n) + f'(x_n - T_2 c_n) \\ &= \left[ \frac{f'(x_n + T_1 c_n)}{x_n - x^* + T_1 c_n} + \frac{f'(x_n - T_2 c_n)}{x_n - x^* - T_2 c_n} \right] (x_n - x^*) \\ &\quad + c_n \left[ T_1 \frac{f'(x_n + T_1 c_n)}{x_n - x^* + T_1 c_n} - T_2 \frac{f'(x_n - T_2 c_n)}{x_n - x^* - T_2 c_n} \right]. \qquad (13) \end{aligned}$$

Now note that, using (F1) and (F2) we have

$$K_0 \leq \left| \frac{f'(x)}{x - x^*} \right| = -\frac{f'(x)}{x - x^*} \leq K_1. \qquad (14)$$

Using (13), we have

$$\begin{aligned} (x_n - x^*) \widehat{\nabla} f(x_n) &= \left[ \frac{f'(x_n + T_1 c_n)}{x_n - x^* + T_1 c_n} + \frac{f'(x_n - T_2 c_n)}{x_n - x^* - T_2 c_n} \right] (x_n - x^*)^2 \\ &\quad + c_n (x_n - x^*) \left[ T_1 \frac{f'(x_n + T_1 c_n)}{x_n - x^* + T_1 c_n} - T_2 \frac{f'(x_n - T_2 c_n)}{x_n - x^* - T_2 c_n} \right] \\ &\leq -2K_0 (x_n - x^*)^2 + K_1 c_n |x_n - x^*|, \qquad (15) \end{aligned}$$

where the inequality uses the fact that $f'(x)/(x - x^*) \leq -K_0$ for any $x \in \mathbb{R}$, and the second term follows from

$$\begin{aligned} \left| T_1 \frac{f'(x_n + T_1 c_n)}{x_n - x^* + T_1 c_n} - T_2 \frac{f'(x_n - T_2 c_n)}{x_n - x^* - T_2 c_n} \right| &\leq \max \left\{ T_1 \left| \frac{f'(x_n + T_1 c_n)}{x_n - x^* + T_1 c_n} \right|, T_2 \left| \frac{f'(x_n - T_2 c_n)}{x_n - x^* - T_2 c_n} \right| \right\} \\ &\leq \max\{T_1 K_1, T_2 K_1\} \\ &\leq K_1. \qquad (16) \end{aligned}$$

Now, using the inequality $|a+b|^r \leq 2^{r-1}(|a|^r + |b|^r)$ with (13), we obtain

$$\begin{aligned}
[\widehat{\nabla} f(x_n)]^2 &\leq 2\left[\frac{f'(x_n + T_1 c_n)}{x_n - x^* + T_1 c_n} + \frac{f'(x_n - T_2 c_n)}{x_n - x^* - T_2 c_n}\right]^2 (x_n - x^*)^2 \\
&\quad + 2c_n^2 \left[T_1 \frac{f'(x_n + T_1 c_n)}{x_n - x^* + T_1 c_n} - T_2 \frac{f'(x_n - T_2 c_n)}{x_n - x^* - T_2 c_n}\right]^2 \\
&\leq 8K_1^2 (x_n - x^*)^2 + 2K_1^2 c_n^2, \qquad (17)
\end{aligned}$$

where the last inequality follows from bounding the first term using (14) and the second term using (16).

**Step 2:** Let $X_n$ be the output of the $n^{th}$ iterate of (1) and let $Y_{2n}, Y_{2n-1}$ be the function observations at points $X_n + c_n$ and $X_n - c_n$ respectively. Note that

$$\mathbb{E}\left(\frac{Y_{2n} - Y_{2n-1}}{c_n} \,\middle|\, X_n\right) = \frac{f(X_n + c_n) - f(X_n - c_n)}{c_n} =: \widehat{\nabla} f(X_n),$$

which together with the bounded variance assumption implies that

$$\mathbb{E}\left[\left(\frac{Y_{2n} - Y_{2n-1}}{c_n}\right)^2 \,\middle|\, X_n\right] = \mathrm{Var}\left(\frac{Y_{2n} - Y_{2n-1}}{c_n} \,\middle|\, X_n\right) + [\widehat{\nabla} f(X_n)]^2 \leq \frac{2\sigma^2}{c_n^2} + [\widehat{\nabla} f(X_n)]^2. \qquad (18)$$

Now, using the above and (1) we have

$$\begin{aligned}
Z_{n+1} &:= (X_{n+1} - x^*)^2 \\
&= \left[X_n - x^* + a_n \left(\frac{Y_{2n} - Y_{2n-1}}{c_n}\right)\right]^2 \qquad (19) \\
&= (X_n - x^*)^2 + 2a_n (X_n - x^*)\left(\frac{Y_{2n} - Y_{2n-1}}{c_n}\right) + a_n^2 \left(\frac{Y_{2n} - Y_{2n-1}}{c_n}\right)^2.
\end{aligned}$$

Taking expectations of both sides conditioned on $X_n$ and using (18) we get

$$\mathbb{E}(Z_{n+1}|X_n) \leq Z_n + 2a_n (X_n - x^*)\widehat{\nabla} f(X_n) + a_n^2 \left(\frac{2\sigma^2}{c_n^2} + [\widehat{\nabla} f(X_n)]^2\right). \qquad (20)$$

Using (15) and (17) we have

$$\mathbb{E}(Z_{n+1}|X_n) \leq Z_n - 4a_n K_0 Z_n + 2K_1 a_n c_n \sqrt{Z_n} + 2\frac{a_n^2}{c_n^2}\sigma^2 + 8K_1^2 a_n^2 Z_n + 2K_1^2 a_n^2 c_n^2. \qquad (21)$$

Finally, taking expectations, using the inequality $\mathbb{E}(\sqrt{Z_n}) \leq \sqrt{\mathbb{E}(Z_n)}$, and setting $b_n := \mathbb{E}(Z_n)$

we get the following recursion:

$$b_{n+1} \leq (1 - 4a_nK_0 + 8K_1^2a_n^2)b_n + 2K_1a_nc_n\sqrt{b_n} + 2\frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2. \tag{22}$$

**Step 3:** Before we start the induction proof, we will derive a crude upper bound on $b_n$ that will be used later. Using $\sqrt{b_n} \leq 1 + b_n$ in (22) we get

$$b_{n+1} \leq b_n(1 - 4a_nK_0 + 8K_1^2a_n^2 + 2K_1a_nc_n) + 2K_1a_nc_n + 2\frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2,$$

which can be expressed more compactly as

$$b_{n+1} \leq b_np_n + q_n, \tag{23}$$

with:

$$\begin{aligned} p_n &:= 1 - 4a_nK_0 + 8K_1^2a_n^2 + 2K_1a_nc_n > 0 \\ q_n &:= 2K_1a_nc_n + 2\frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2. \end{aligned}$$

Note that since $2K_1a_nc_n > 0$ by (S3), we have $p_n \geq 1-4a_nK_0+8K_1^2a_n^2$ which is a quadratic equation in $a_n$ with positive leading coefficient, and $0 < K_0 < K_1$ ensures it has negative discriminant, hence $p_n > 0$.

Solving recursion (23), we get that for all $n$

$$b_n \leq b_1\prod_{i=1}^{n} p_i + \sum_{i=2}^{n-1} q_i \prod_{j=i+1}^{n} p_j + q_n := B_n \tag{24}$$

which provides a crude upper bound on the MSE at the $n^{th}$ step of the algorithm.

Put

$$n_0 := \sup\{n \geq 1 : (8K_1^2 - 4K_0A)a_n + 8K_1^2Aa_n^2 \geq 4K_0 - A\} + 1, \tag{25}$$

and set $n_0 = 1$ if $(8K_1^2 - 4K_0A)a_n + 8K_1^2Aa_n^2 < 4K_0 - A$ for all $n$. Since we assume $A < 4K_0$, we have $n_0 < \infty$ because $a_n \to 0$ as $n \to \infty$ (assumption (S3)). Also, note that by (25)

$$\zeta := -\sup\{A - 4K_0 + (8K_1^2 - 4K_0A)a_n + 8K_1^2Aa_n^2 : n \geq n_0\} > 0. \tag{26}$$

**Step 4:** Now we will carry out the induction part of the proof.

Case (i): Suppose $c_n^4/a_n \leq \tau_1$, for all $n \geq 1$. We will first show that $b_{n+1} \leq C_1a_n/c_n^2$ for all $n \geq n_0$ and some finite positive constant $C_1$. First, for $n = n_0$ suppose $C_1$ is chosen large enough

to ensure $C_1 \geq B_{n_0+1}c_{n_0}^2/a_{n_0} \geq b_{n_0+1}c_{n_0}^2/a_{n_0}$. Now fix $n > n_0$ and suppose $b_{k+1} \leq C_1 a_k/c_k^2$ for all $n_0 \leq k \leq n-1$. We need to show that $b_{n+1} \leq C_1 a_n/c_n^2$. Using inequality (22) and the induction hypothesis we have

$$\begin{aligned} b_{n+1} &\leq (1 - 4a_nK_0 + 8K_1^2a_n^2)C_1\frac{a_{n-1}}{c_{n-1}^2} + 2K_1a_nc_n\sqrt{C_1}\frac{\sqrt{a_{n-1}}}{c_{n-1}} + 2\frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2 \\ &\leq C_1\frac{a_n}{c_n^2}(1 + Aa_n) - 4K_0C_1\frac{a_n^2}{c_n^2}(1 + Aa_n) + 8K_1^2C_1\frac{a_n^3}{c_n^2}(1 + Aa_n) \\ &+ 2K_1\sqrt{C_1}a_n^{3/2}(1 + \frac{A}{2}a_n) + 2\frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2, \end{aligned}$$

where for the second inequality we have used condition (S1) and the inequality $\sqrt{1 + Aa_n} \leq 1 + Aa_n/2$. Rearranging terms we get

$$\begin{aligned} b_{n+1} &\leq C_1\frac{a_n}{c_n^2} + \frac{a_n^2}{c_n^2}\Big\{ C_1[A - 4K_0 + (8K_1^2 - 4K_0A)a_n + 8K_1^2Aa_n^2] \\ &+ 2\sqrt{C_1}K_1(\frac{c_n^2}{\sqrt{a_n}} + \frac{A}{2}\sqrt{a_n}c_n^2) + 2\sigma^2 + 2K_1^2c_n^4 \Big\}. \end{aligned}$$

Letting $\nu$ and $\kappa$ denote the upper bounds on the $\{a_n\}$ and $\{c_n\}$ sequences, respectively, and using $c_n^2/\sqrt{a_n} \leq \sqrt{\tau_1}$, (26) gives:

$$b_{n+1} \leq C_1\frac{a_n}{c_n^2} + \frac{a_n^2}{c_n^2}\left[-C_1\zeta + 2\sqrt{C_1}K_1(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2) + 2\sigma^2 + 2K_1^2\kappa^4\right]. \tag{27}$$

Now, if we can show that for some finite positive constant $C_1$,

$$-C_1\zeta + 2\sqrt{C_1}K_1(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2) + 2\sigma^2 + 2K_1^2\kappa^4 \leq 0, \tag{28}$$

then the induction proof would be complete. Viewing this as a quadratic in $\sqrt{C_1}$, we first observe that the leading coefficient is negative, by (26). It follows that this quadratic admits a solution, in particular, solving for the positive root and using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we have $b_{n+1} \leq C_1 a_n/c_n^2$ for all $n \geq n_0$ with any choice of $C_1$ satisfying

$$C_1 \geq \max\left\{\left[\frac{2K_1(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2)}{\zeta} + \sqrt{\frac{2\sigma^2 + 2K_1^2\kappa^4}{\zeta}}\right]^2, \frac{c_{n_0}^2}{a_{n_0}}B_{n_0+1}\right\}. \tag{29}$$

Finally let us modify the constant $C_1$ so that the result holds for all $n \geq 1$. This requires a

simple modification in (29), and using $b_n \leq B_n$ can be done by setting

$$C_1 = \max\left\{ \left[ \frac{2K_1(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2)}{\zeta} + \sqrt{\frac{2\sigma^2 + 2K_1^2\kappa^4}{\zeta}} \right]^2, \max_{1\leq n\leq n_0} \left\{ \frac{c_n^2}{a_n} B_{n+1} \right\} \right\}. \tag{30}$$

Case (ii): Suppose $c_n^4/a_n \geq \tau_2$, for all $n \geq 1$. Using similar steps to those in the proof of case (i), we will first show that $b_{n+1} \leq C_2 c_n^2$ for all $n \geq n_0$ for some finite positive constant $C_2$. First, for $n = n_0$ suppose $C_2$ is chosen large enough to assure $C_2 \geq B_{n_0+1}/c_{n_0}^2 \geq b_{n_0+1}/c_{n_0}^2$. Now suppose we have $b_{k+1} \leq C_2 c_k^2$ for all $n_0 \leq k \leq n-1$. We need to prove $b_{n+1} \leq C_2 c_n^2$. Using inequality (22) and the induction hypothesis we have

$$\begin{aligned} b_{n+1} &\leq (1 - 4a_nK_0 + 8K_1^2a_n^2)C_2c_{n-1}^2 + 2K_1a_nc_n\sqrt{C_2}c_{n-1} + 2\frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2 \\ &\leq C_2c_n^2(1 + Aa_n) - 4K_0C_2a_nc_n^2(1 + Aa_n) + 8K_1^2C_2a_n^2c_n^2(1 + Aa_n) \\ &+ 2K_1\sqrt{C_2}a_nc_n^2(1 + \frac{A}{2}a_n) + 2\frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2. \end{aligned}$$

where for the second inequality we have used (S2) with the inequality $\sqrt{1 + Aa_n} \leq 1 + Aa_n/2$. Rearranging terms we get

$$\begin{aligned} b_{n+1} &\leq C_2c_n^2 + a_nc_n^2 \Big\{ C_2[A - 4K_0 + (8K_1^2 - 4K_0A)a_n + 8K_1^2Aa_n^2] \\ &+ 2\sqrt{C_2}K_1(1 + \frac{A}{2}a_n) + 2\sigma^2\frac{a_n}{c_n^4} + 2K_1^2a_n \Big\}. \end{aligned}$$

Using $a_n \leq \nu$, (26) and the assumption that $a_n/c_n^4 \leq 1/\tau_2$, we get

$$b_{n+1} \leq C_2c_n^2 + a_nc_n^2\left[ -C_2\zeta + 2\sqrt{C_2}K_1(1 + \frac{A\nu}{2}) + \frac{2\sigma^2}{\tau_2} + 2K_1^2\nu \right]. \tag{31}$$

Similar to the first case, we need

$$-C_2\zeta + 2\sqrt{C_2}K_1(1 + \frac{A\nu}{2}) + \frac{2\sigma^2}{\tau_2} + 2K_1^2\nu \leq 0$$

for a suitable choice of $C_2$. Using the same argument as before, we have $b_{n+1} \leq C_2c_n^2$ for all $n \geq n_0$ with any $C_2$ satisfying

$$C_2 \geq \max\left\{ \left[ \frac{2K_1(1 + \frac{A\nu}{2})}{\zeta} + \sqrt{\frac{2\sigma^2/\tau_2 + 2K_1^2\nu}{\zeta}} \right]^2, \frac{1}{c_{n_0}^2}B_{n_0+1} \right\}. \tag{32}$$

Setting

$$C_2 = \max\left\{ \left[ \frac{2K_1(1+\frac{A\nu}{2})}{\zeta} + \sqrt{\frac{2\sigma^2/\tau_2 + 2K_1^2\nu}{\zeta}} \right]^2, \max_{1\le n\le n_0}\left\{ \frac{1}{c_n^2} B_{n+1} \right\} \right\}. \tag{33}$$

we get the result for all $n \geq 1$ and this completes the proof. ■

## A.2 Proof of Proposition 1

First, we claim that the optimal rate of convergence is achieved with sequences $\{a_n\}$ and $\{c_n\}$ that satisfy $c_n^4/a_n = \tau$. To see this note that if $c_n^4/a_n < \tau$, then we are in case (i) of Theorem 1 and the rate of convergence is $a_n/c_n^2$. But if we increase $c_n$ or decrease $a_n$ until we get $c_n^4/a_n = \tau$, we achieve a tighter bound. The same line of argument applies when $c_n^4/a_n > \tau$. Hence the best possible bound in (2) is of order $\sqrt{a_n}$. Thus, once we specify the $\{a_n\}$ sequence, we set $\{c_n\}$ such that $c_n^4/a_n = \tau$.

Next we show that $a_n = O(1/n)$ is the optimal order of magnitude for the $\{a_n\}$ sequence, in the sense that this choice yields the fastest convergence to zero of the MSE among those sequences satisfying assumptions (S1)-(S4). Now, clearly $a_n = \theta_a/n$ and $c_n = \theta_c/n^{1/4}$ satisfy assumptions (S1)-(S4). Suppose, towards a contradiction, that $\{a_n\}$ is of lower order than $1/n$, i.e., suppose $a_n = \theta_a s_n/n$ for some finite positive sequence $\{s_n\}$ such that $s_n \to 0$ as $n \to \infty$ and $\{a_n\}$ is non-increasing.

We first observe that since $\{a_n\}$ is non-increasing, i.e., $a_{n+1} \leq a_n$, we have $s_{n+1} \leq s_n(n+1)/n$. Using this, we have

$$\frac{s_{n+1}^2}{n+1} \leq \frac{s_n^2}{n}\left(1+\frac{1}{n}\right) \leq 2\frac{s_n^2}{n}. \tag{34}$$

Now, note that (S1) and (S2) imply $a_n \leq a_{n+1}(1+\overline{A}a_{n+1})$ for some constant $\overline{A}$. Using this, we have $s_n/n \leq s_{n+1}/(n+1) + \overline{A}\theta_a s_{n+1}^2/(n+1)^2$, which implies

$$\begin{aligned} s_{n+1} &\geq s_n\left(1+\frac{1}{n}\right) - \overline{A}\theta_a \frac{s_{n+1}^2}{n+1} \\ &\geq s_n\left(1+\frac{1}{n} - 2\overline{A}\theta_a\frac{s_n}{n}\right), \end{aligned} \tag{35}$$

where we used (34) for the second inequality. Since $s_n \to 0$ as $n \to \infty$, we have that $2\overline{A}\theta_a s_n \leq 1/2$ for $n$ sufficiently large. Using this in (35), we get $s_{n+1} \geq s_n(1+1/(2n))$ for all $n$ sufficiently large. But this implies $s_n \to \infty$, which contradicts the assumption $s_n \to 0$ as $n \to \infty$. Therefore the optimal choice is $a_n = \theta_a/n$, and hence $c_n = \theta_c/n^{1/4}$ for positive constants $\theta_a$ and $\theta_c$.

The last step is to find the restrictions on $\theta_a$ and $\theta_c$ to ensure that the condition $A < 4K_0$ holds. If we substitute these sequences in (S1) or (S2), after some algebra, we get

$$A \geq \frac{n+1}{\theta_a}\left(\sqrt{\frac{n+1}{n}} - 1\right).$$

Combining this with $A < 4K_0$, we get

$$\theta_a > \frac{1}{4K_0}\left[(n+1)\left(\sqrt{\frac{n+1}{n}} - 1\right)\right] \geq \frac{\sqrt{2}-1}{2K_0},$$

since the term in square brackets is maximized when $n = 1$. ■

### A.3 Proof of Proposition 2

**Step 1:** Without loss of generality, we can assume $c_n \leq C_0$ for all $n \geq 1$ since we can scale down the sequence otherwise. Using (F3) and $\widehat{\nabla} f(x) = (f(x_n + c_n) - f(x_n - c_n))/c_n$ we have

$$(x - x^*)\widehat{\nabla} f(x) \leq -K_0(x - x^*)^2. \tag{36}$$

Squaring the terms in (F3), we obtain

$$[\widehat{\nabla} f(x)]^2 \leq K_1^2(x - x^*)^2. \tag{37}$$

**Step 2:** We now derive a real number recursion for $b_n = \mathbb{E}(Z_n)$ using the same ideas as in Step 2 of the proof of Theorem 1. Using (20) and taking expectations of both sides and applying the bounds (36) and (37), denoting $b_n = \mathbb{E}(Z_n)$, we get

$$\begin{aligned} b_{n+1} &\leq b_n - 2a_nK_0b_n + 2\frac{a_n^2}{c_n^2}\sigma^2 + K_1^2a_n^2b_n \\ &= b_n(1 - 2a_nK_0 + K_1^2a_n^2) + 2\frac{a_n^2}{c_n^2}\sigma^2. \end{aligned} \tag{38}$$

**Step 3:** In the third step of the proof of Theorem 1, we established a bound for sequences satisfying $b_{n+1} \leq b_np_n + q_n$. Using this, we have $b_n \leq B_n$, where $B_n$ is defined in (24)and $p_n := 1 - 2a_nK_0 + K_1^2a_n^2 > 0$, $q_n := 2\sigma^2a_n^2/c_n^2$.

Now define

$$n_0' := \sup\{n \geq 1 : (K_1^2 - 2AK_0)a_n + K_1^2Aa_n^2 \geq 2K_0 - A\} + 1 \tag{39}$$

and set $n_0' = 1$ if $(K_1^2 - 2AK_0)a_n + K_1^2Aa_n^2 < 2K_0 - A$ for all $n \geq 1$. Using $A < 2K_0$, and since it

is assumed in (S3) that $a_n \to 0$ as $n \to \infty$ we have $n_0' < \infty$. We note that by (39)

$$\zeta := -\sup\{A - 2K_0 + (K_1^2 - 2AK_0)a_n + K_1^2 A a_n^2 : n \geq n_0'\} > 0. \tag{40}$$

**Step 4:** We will again use induction to complete the proof. For $n = n_0'$ suppose $C$ is chosen large enough to ensure $C \geq B_{n_0'+1} c_{n_0'}^2 / a_{n_0'} \geq b_{n_0'+1} c_{n_0'}^2 / a_{n_0'}$. Now suppose $b_{k+1} \leq C a_k / c_k^2$ for all $n_0' \leq k \leq n-1$. We need to show that $b_{n+1} \leq C a_n / c_n^2$. Using (38) and the induction hypothesis we have

$$\begin{aligned} b_{n+1} &\leq (1 - 2a_n K_0 + K_1^2 a_n^2) C \frac{a_{n-1}}{c_{n-1}^2} + 2\frac{a_n^2}{c_n^2}\sigma^2 \\ &\leq C\frac{a_n}{c_n^2}(1 + Aa_n) - 2K_0 C \frac{a_n^2}{c_n^2}(1 + Aa_n) + K_1^2 C \frac{a_n^3}{c_n^2}(1 + Aa_n) + 2\frac{a_n^2}{c_n^2}\sigma^2, \end{aligned}$$

where for the second inequality we have used condition (S1). Rearranging terms we get

$$b_{n+1} \leq C\frac{a_n}{c_n^2} + \frac{a_n^2}{c_n^2}\left\{ C(A - 2K_0 + (K_1^2 - 2K_0 A)a_n + K_1^2 A a_n^2) + 2\sigma^2 \right\}. \tag{41}$$

Using (40) gives

$$b_{n+1} \leq C\frac{a_n}{c_n^2} + \frac{a_n^2}{c_n^2}(-C\zeta + 2\sigma^2). \tag{42}$$

Using a similar argument to the one in proof of Theorem 1, with

$$C = \max\left\{ \frac{2\sigma^2}{\zeta}, \max_{1 \leq n \leq n_0'} \left\{ \frac{c_n^2}{a_n} B_n + 1 \right\} \right\}. \tag{43}$$

we have $b_{n+1} \leq C a_n / c_n^2$ for all $n \geq 1$ and this completes the proof. ■

## A.4 Proof of Proposition 3

First note that with the choice of sequences and with $\theta_a > (\sqrt{2} - 1)/K_0$, we satisfy all the conditions of Proposition 2 (the inequality is verified at the end of the proof). Using this, the rate of convergence is $a_n / c_n^2$. To obtain the optimal rate of convergence, we should choose $a_n / c_n^2$ as small as possible such that it satisfies the assumptions in the proposition. Using assumption (S1), and $c_n \leq \kappa$ for all $n \geq 1$, we get

$$\frac{a_n}{c_n^2} \leq \frac{a_{n+1}}{c_{n+1}^2}(1 + Aa_{n+1}) \leq \frac{a_{n+1}}{c_{n+1}^2}(1 + A\kappa^2 \frac{a_{n+1}}{c_{n+1}^2}). \tag{44}$$

Substituting $d_n = a_n / c_n^2$ and $D = A\kappa^2$ in (44), to complete the proof, we need to find the non-increasing sequence $\{d_n\}$ satisfying $d_n \leq d_{n+1}(1 + D d_{n+1})$ which converges to zero as fast as possible.

But in the proof of Proposition 1, we showed that under these condition, the best choice of $\{d_n\}$, in the sense of minimizing the order of the MSE, is $d_n = \theta_d/n$ for some finite positive constant $\theta_d$. This can be achieved by choosing $a_n = \theta_a/n$ and $c_n = \theta_c$ for some finite positive constants $\theta_a$ and $\theta_c$. As we did in the proof of Proposition 1, the last step is to translate the requirement $A < 2K_0$ into a condition on $\theta_a$. Substituting the sequences in assumption (S1) gives the condition $A \geq [(n+1)/n]/\theta_a$. With $A < 2K_0$, this requires $\theta_a > [(n+1)/n]/(2K_0)$ which completes the proof since the term in the square brackets is maximized when $n = 1$. ■

# B Multidimensional KW algorithm

A multidimensional version of the KW algorithm was introduced by Blum [4] and uses a finite difference gradient approximation in each direction using independent observations and the same $\{c_n\}$ sequence. Suppose $f : \mathbb{R}^d \to \mathbb{R}$, and let $\{e_1, \ldots, e_d\}$ is the standard basis in $\mathbb{R}^d$. Define $Y_{x,c} = [(Y_{x+ce_1} - Y_x), \ldots, (Y_{x+ce_d} - Y_x)]$ where $Y_x$ is an observation of the function value at point $x \in \mathbb{R}^d$. Blum [4] defines the stochastic approximation scheme as follows:

$$X^{n+1} = X^n + a_n \frac{Y^n}{c_n} \quad n = 1, 2, \ldots \tag{45}$$

where $Y^n$ is equal in distribution to $Y_{X^n,c_n}$. As in the one-dimensional case, we assume

$$\sigma^2 := \sup_{x \in \mathbb{R}} \mathrm{Var}[Y_x] < \infty.$$

For this set up, we introduce the following obvious generalizations of (F1) and (F2):

(F1') There exist finite positive constants $K_0$ and $K_1$ such that

$$K_0|x_i - x_i^*| \leq \left| \frac{\partial f(x)}{\partial x_i} \right| \leq K_1|x_i - x_i^*|,$$

for all $x \in \mathbb{R}^d$ and for all $i = 1, \ldots, d$ where $x_i$ denotes the $i^{th}$ coordinate of $x$, and $\partial f(x)/\partial x_i$ the respective partial derivative.

(F2') $\partial f(x)/\partial x_i \cdot (x_i - x_i^*) < 0$ for all $x \in \mathbb{R}^d \setminus \{x^*\}$ and all $i = 1, \ldots, d$.

**Corollary 1** *Let $\{X^n\}$ be generated by the recursion (45) using $\{a_n\}$ and $\{c_n\}$ satisfying (S1)-(S4) with $A < 2K_0$. Then under assumptions (F1') and (F2'),*

$$\mathbb{E}\|X^{n+1} - x^*\|^2 \leq \begin{cases} C_1'' a_n/c_n^2 & \text{if} \quad c_n^4 \leq \tau_1 a_n \\ C_2'' c_n^2 & \text{if} \quad c_n^4 \geq \tau_2 a_n \end{cases} \tag{46}$$

*for all $n \geq 1$, and some finite positive constants $C_1''$, $C_2''$.*

**Remark 13** *Note that the assumption $A < 4K_0$ of Theorem 1 is replaced by $A < 2K_0$ in the multidimensional KW-algorithm. This is due to using a one-sided finite difference approximation to the gradient in the multidimensional case versus a two-sided approximation in one dimension.*

**Proof.** The proof follows that of Theorem 1.

**Step 1:** Fix $\{a_n\}$ and $\{c_n\}$ as in the statement of the theorem. Setting $f_i'(y) := \partial f(y)/\partial x_i$ and using Taylor expansion, for positive integer $n$, $x_n \in \mathbb{R}^d$ and $c \in \mathbb{R}_+$, there exist $T = [T_1, \ldots, T_d]$ with $0 \leq T_i \leq 1$ for all $i = 1, \ldots, d$, such that $f(x^n + ce_i) = f(x^n) + cf_i'(x^n + T_i c e_i)$, since $e_i$ has all zero entries except its $i^{th}$ coordinate.

Therefore, we have

$$\begin{aligned} \widehat{\nabla} f_i(x) &:= \frac{f(x + c_n e_i) - f(x)}{c_n} = f_i'(x + T_i c_n e_i) && (47) \\ &= \frac{f_i'(x + T_i c_n e_i)}{x_i - x_i^* + T_i c_n}(x_i - x_i^*) + c_n T_i \frac{f'(x + T_i c_n)}{x_i - x_i^* + T_i c_n}. && (48) \end{aligned}$$

Now, using (F1') and (F2') we have

$$K_0 \leq \left| \frac{f_i'(x)}{x_i - x_i^*} \right| = -\frac{f_i'(x)}{x_i - x_i^*} \leq K_1. \tag{49}$$

which ensures (48) is well defined, i.e., the denominator is non-zero. Using (48), we have

$$\begin{aligned} \sum_{i=1}^{d} (x_i - x_i^*) \widehat{\nabla} f_i(x) &= \sum_{i=1}^{d} \left[ \frac{f_i'(x + T_i c_n e_i)}{x_i - x_i^* + T_i c_n}(x_i - x_i^*)^2 + c_n (x_i - x_i^*) T_i \frac{f'(x + T_i c_n)}{x_i - x_i^* + T_i c_n} \right] \\ &\leq -K_0 \|x - x^*\|^2 + K_1 \sqrt{d} c_n \|x - x^*\|, && (50) \end{aligned}$$

where $\|\cdot\|$ denotes the Euclidean norm and the inequality follows from $f_i'(x)/(x_i - x_i^*) \leq -K_0$ for any $x \in \mathbb{R}^d$.

Now, using the inequality $|a+b|^r \leq 2^{r-1}(|a|^r + |b|^r)$ with (48), and defining $\widehat{\nabla} f(x) := (\widehat{\nabla} f_1(x), \ldots, \widehat{\nabla} f_d(x))$, we obtain

$$\begin{aligned} \|\widehat{\nabla} f(x)\|^2 &= \sum_{i=i}^{d} [\widehat{\nabla} f_i(x)]^2 \\ &\leq 2 \sum_{i=i}^{d} \left[ \frac{f_i'(x + T_i c_n e_i)}{x_i - x_i^* + T_i c_n} \right]^2 (x_i - x_i^*)^2 + 2c_n^2 \sum_{i=i}^{d} T_i^2 \left[ \frac{f'(x + T_i c_n)}{x_i - x_i^* + T_i c_n} \right]^2 \\ &\leq 2K_1^2 \|x - x^*\|^2 + 2dK_1^2 c_n^2. && (51) \end{aligned}$$

where the last inequality follows from (49).

**Step 2:** Let $X^n$ be the output of the $n^{th}$ iterate of (45). Note that

$$\mathbb{E}\left(\frac{Y_{X^n+c_ne_i}-Y_{X^n}}{c_n}\,\Big|\; X^n\right)=\widehat{\nabla}f_i(X^n),$$

which together with bounded variance assumption implies that

$$\begin{aligned}\mathbb{E}\left[\left\|\frac{Y^n}{c_n}\right\|^2\,\Big|\; X^n\right] &= \sum_{i=1}^{d}\mathrm{Var}\left(\frac{Y_{X^n+c_ne_i}-Y_{X^n}}{c_n}\,\Big|\; X^n\right)+\sum_{i=1}^{d}\left[\widehat{\nabla}f_i(X^n)\right]^2\\ &\leq \frac{2d\sigma^2}{c_n^2}+\|\widehat{\nabla}f(X^n)\|^2. \end{aligned}\tag{52}$$

Now, using the above and (45) we have

$$\begin{aligned}Z^{n+1} &:= \|X^{n+1}-x^*\|^2=\left\|X^n-x^*+a_n\frac{Y^n}{c_n}\right\|^2\\ &= \|X^n-x^*\|^2+2a_n\sum_{i=1}^{d}(X_i^n-x_i^*)\frac{Y_i^n}{c_n}+a_n^2\left\|\frac{Y^n}{c_n}\right\|^2.\end{aligned}$$

Taking expectations of both sides conditioned on $X^n$ and using (52) we get

$$\mathbb{E}(Z^{n+1}|X^n)\leq Z^n+2a_n\sum_{i=1}^{d}(X_i^n-x_i^*)\widehat{\nabla}f_i(X^n)+a_n^2\left(\frac{2d\sigma^2}{c_n^2}+\|\widehat{\nabla}f(X^n)\|^2\right).$$

Using bounds (50) and (51) we have

$$\mathbb{E}(Z^{n+1}|X^n)\leq Z^n-2a_nK_0Z^n+2K_1\sqrt{d}a_nc_n\sqrt{Z^n}+2d\frac{a_n^2}{c_n^2}\sigma^2+2K_1^2a_n^2Z^n+2dK_1^2a_n^2c_n^2.\tag{53}$$

Finally, taking expectations, using Jensen's inequality, and setting $b_n:=\mathbb{E}(Z^n)$ we get the following recursion:

$$b_{n+1}\leq(1-2a_nK_0+2K_1^2a_n^2)b_n+2K_1\sqrt{d}a_nc_n\sqrt{b_n}+2d\frac{a_n^2}{c_n^2}\sigma^2+2dK_1^2a_n^2c_n^2.\tag{54}$$

Since (54) is of the same form as (22) in proof of Theorem 1, the rest of the proof is a step-by-step replication of the induction-based proof of Theorem 1. We omit the details. ■

## References


[1] S. Andradóttir. A stochastic approximation algorithm with varying bounds. *Operations Research*, 43(6):1037–1048, 1995.

[2] S. Andradóttir. A scaled stochastic approximation algorithm. *Management Science*, 42(4):475–498, 1996.

[3] J.R. Blum. Approximation methods which converge with probability one. *The Annals of Mathematical Statistics*, 25(2):382–386, 1954.

[4] J.R. Blum. Multidimensional stochastic approximation methods. *The Annals of Mathematical Statistics*, 25(4):737–744, 1954.

[5] D.L. Burkholder. On a class of stochastic approximation processes. *The Annals of Mathematical Statistics*, 27(4):1044–1059, 1956.

[6] K.L. Chung. On a stochastic approximation method. *The Annals of Mathematical Statistics*, 25(3):463–483, 1954.

[7] C. Derman. An application of Chung's lemma to the Kiefer-Wolfowitz stochastic approximation procedure. *The Annals of Mathematical Statistics*, 27(2):532–536, 1956.

[8] J. Dippon and J. Renz. Weighted means in stochastic approximation of minima. *SIAM J. Control Optim.*, 35(5):1811–1827, 1997.

[9] V. Dupac. O Kiefer-Wolfowitzově approximační methodě. *Časopis pro Pěstování Matematiky*, 82:47–75, 1957.

[10] V. Fabian. Stochastic approximation of minima with improved asymptotic speed. *The Annals of Mathematical Statistics*, 38(1):191–200, 1967.

[11] A. Juditsky, G. Lan, A. Nemirovski, and A. Shapiro. Stochastic approximation approach to stochastic programming. *Submitted to SIAM Journal on Optimization*, 2007.

[12] H. Kesten. Accelerated stochastic approximation. *The Annals of Mathematical Statistics*, 29(1):41–59, 1958.

[13] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[14] H.J. Kushner and G.G. Yin. *Stochastic approximation and recursive algorithms and applications*. Springer-Verlag, New York, 2003.

[15] T.L. Lai. Stochastic approximation. *Annals of Statistics*, 31(2):391–406, 2003.

[16] A. Mokkadem and M. Pelletier. A companion for the Kiefer-Wolfowitz-Blum stochastic approximation algorithm. *The Annals of Statistics*, 35(4):1749–1772, 2007.

[17] B.T. Polyak. New stochastic approximation type procedures. *Automat. i Telemekh*, 7:98–107, 1990.

[18] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[19] J. Sacks. Asymptotic distribution of stochastic approximation procedures. *The Annals of Mathematical Statistics*, 29(2):373–405, 1958.

[20] A. Tsybakov and B.T. Polyak. Optimal order of accuracy of search algorithms in stochastic optimization. *Problemy Peredachi Informatsii*, 26(2):45–53, 1990.