1. Introduction

Background and Motivation. The term stochastic approximation refers to a broad class of optimization problems in which function values can be computed only in the presence of noise. Representative examples include stochastic estimation of a zero crossing, first introduced in the work of Robbins and Monro (1951), and stochastic estimation of the point of maximum, first studied by Kiefer and Wolfowitz (1952). 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 (2003) and the book by Kushner and Yin (2003).

A natural setting in which one encounters the need for stochastic approximation algorithms is simulation-based optimization. Here it is possible to evaluate a function only by means of simulation, and the observation noise is a direct consequence of the sample generating scheme; see, for example, Andradóttir (1995, 1996) for further discussion.

For concreteness we focus in this paper 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 (RM) type algorithms; more specific commentary will be given in §2 and the electronic companion to this paper, which is part of the online version that can be found at http://or.journal.informs.org/. In particular, we consider the following stochastic approximation scheme first studied by Kiefer and Wolfowitz (1952):

\[ X_{n+1} = X_n + a_n \left( \frac{\hat{f}(X_n + c_n) - \hat{f}(X_n - c_n)}{c_n} \right), \quad n = 1, 2, \ldots \]  

Here \( X_1 \) is the initial condition (either deterministic or random); \( \{a_n\} \) and \( \{c_n\} \) are two real-valued, deterministic tuning sequences; and \( \hat{f}(X_n + c_n) \) and \( \hat{f}(X_n - c_n) \) are drawn according to conditional distribution functions \( H(y \mid X_n + c_n) \) and \( H(y \mid X_n - c_n) \), which have uniformly bounded second moments. Assuming the regression function \( f(x) := \int y \, dH(y \mid x) \) admits a unique point of maximum and is strongly concave, Kiefer and Wolfowitz (1952) 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:

\[ \text{(KW1)} \quad c_n \to 0 \quad \text{as} \quad n \to \infty; \]
(KW2) $\sum_{n=1}^{\infty} a_n = \infty$;

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

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

Shortly after the publication of the Kiefer and Wolfowitz (KW) algorithm, Blum (1954a) established that condition (KW4) is not necessary for convergence, leaving conditions (KW1)–(KW3), which have been imposed in almost all subsequent papers published on the subject (cf. Kushner and Yin 2003, §5.3.3 for a discussion of more general convergence conditions albeit in a more restricted setting). 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 $X_i$ 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 major focus in the literature has been establishing bounds on the mean-squared error (MSE) $E[|X_n - x^*|^2]$, 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., Derman (1956), Dupac (1957), Fabian (1967), Tsybakov and Polyak (1990). A common thread in these papers is that they all rely on a key lemma by Chung (1954) that 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., Burkholder 1956, Sacks 1958, and more recently Mokkadem and Pelletier 2007 as well as references therein).

At a more practical level, the KW algorithm, theoretical convergence guarantees notwithstanding, has often been noted 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 (1958), 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 (1996) observed divergence of the KW algorithm when applied to functions that 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 the function, which could lead to a degraded rate of convergence; see Nemirovski et al. (2009) for a simple example of this phenomenon. Ruppert (1988), Polyak (1990), and Polyak and Juditsky (1992) introduced the idea of iterate averaging to tackle this issue and proved that it guarantees asymptotically optimal convergence rates. Dippon and Renz (1997) use the same idea to propose a weighted averaging scheme specifically for the KW algorithm; 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 (1952) 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) that 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 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 (1996); 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 might end up oscillating back and forth between the boundaries of the truncation interval (see further discussion in §3). Andradóttir (1995) proposed an algorithm that adaptively determines the truncation interval but still points to the oscillatory behavior as an open problem (see also Chen et al. 1999). Finally, poor performance is also observed when function evaluations tend to be “too noisy,” degrading the quality of the gradient estimate (see Vaidya and Bhatnagar 2006, who propose to replace the gradient estimate with its sign in order to mitigate this effect).

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 simpler and more rudimentary than most extant methods that rely on martingale arguments or tools from weak convergence (cf. Kushner and Yin 2003) and at the same time yields general bounds that hold under broad assumptions on the tuning sequences; see Theorem 1. Our assumptions allow for more general sequences than the ones typically found in the literature (see, for example, Dippon and Spall 1992), and cover cases in which the MSE converges yet the sequences violate necessary conditions for almost sure convergence of the algorithm as laid out, for example, in Chen et al. (1999). The proof technique can be easily applied also to multidimensional settings (e.g., the one in Blum 1954b), randomized modifications of KW
(e.g., the simultaneous perturbation stochastic approximation (SPSA) procedure of Spall 1992), and root finding variants of the Robbins-Monro type; see further commentary following Theorem 1 and the electronic companion. The bounds demonstrate that assumption (KW3) is in fact not necessary for the MSE to converge to zero (see §2.2.2), and at the same time allow us to deduce the optimal choice of tuning 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 property is derived from minimizing the order of our general MSE bounds (see Propositions 1 and 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.

Building on qualitative insights and intuition gleaned from our proofs, we present an adaptive version of the KW and at the same time allow us to deduce the optimal choice of tuning 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 property is derived from minimizing the order of our general MSE bounds (see Propositions 1 and 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.

Remark 1 (Observation Noise). The setting we treat in this paper, requiring the unknown function to be a conditional expectation, namely \(f(x) := \int y dH(y \mid x)\), with bounded variance as in (2), allows for certain dependencies in the observations (e.g., common random numbers in gradient estimation), nonhomogeneous noise and nonadditive noise structure. A common setting for stochastic approximation is one where, conditioned on \(x\), \(\hat{f}(x) = f(x) + e\), where \(e\) is a sequence of independent and identically distributed (i.i.d.) random variables with mean zero and finite variance bounded by \(\sigma^2\). In the additive noise setting, requiring the unknown function to be a conditional expectation essentially restricts the noise to be a martingale difference sequence (which many books and papers on the topic of stochastic approximation take as a primitive assumption).

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 2 (Objective Function). Assumptions (F1) and (F2) are identical to those found in most of the literature and will be used in Theorem 1; cf. Dupac (1957) and Wasan (1969). 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 positive constants \(A\), \(\tau_1\), and \(\tau_2\) satisfy

\[
\begin{align*}
(S1) \quad a_n/c_n^2 &\leq (a_{n+1}/c_{n+1}^2)(1 + Aa_n), \\
(S2) \quad c_n^2 &\leq c_{n+1}^2(1 + Aa_{n+1}) \quad \text{for all } n \geq 1, \\
(S3) \quad a_n &\to 0 \text{ as } n \to \infty, \\
(S4) \quad (i) \quad c_n^4/a_n \leq \tau_1 \text{ or } (ii) \quad c_n^4/a_n \geq \tau_2, \quad \text{for all } n \geq 1.
\end{align*}
\]

Remark 3 (Tuning Sequences). The sequences \(a_n = \theta_n/n^\alpha\) and \(c_n = \theta_n/n^\beta\) for \(0 < a < 1\) and \(\alpha \geq 0\) 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 \(a_n = \theta_n/n\) and \(c_n = \theta_n/n\) for \(0 < a < 1\) and \(\alpha \geq 0\) satisfy (S1)–(S4) with \(\theta_n = \theta_n\) and \(\theta_n\) 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),

\[
\begin{align*}
\mathbb{E}(X_{n+1} - x^*)^2 &\leq \\
&\begin{cases} 
C_1a_n/c_n^2 & \text{if } c_n^4 \leq \tau_1a_n \\
C_2c_n^2 & \text{if } c_n^4 \geq \tau_2a_n
\end{cases}
\end{align*}
\]

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

\[
\sigma^2 := \sup_{x \in \mathbb{R}} \text{Var}[\hat{f}(X_n + c_n) - \hat{f}(X_n - c_n) \mid X_n = x] < \infty. \quad (2)
\]
for all \( n \geq 1 \), where \( C_1 \) and \( C_2 \) are finite positive constants identified explicitly in (EC.20) and (EC.23), respectively.

**Proof Outline.** We only sketch the key ideas here; the full proof is given in §EC.1 of the electronic companion. First, using assumptions (F1) and (F2) we derive bounds on the finite difference approximation of the gradient; see (EC.4) and (EC.6). 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

\[
b_{n+1} \leq (1 - 4a_nK_0 + 8K_1^2a_n^2)b_n + 2K_1a_nc_n\sqrt{b_n} + 2a_n^2\sigma^2 + 2K_2a_n^2c_n^2,
\]

(4)

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

Now because \( a_n \to 0 \) as \( n \to \infty \), \((1 - 4a_nK_0 + 8K_1^2a_n^2) < 1\) holds for all \( n \) suitably large, and we eventually have a contraction in recursion (4). 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 \( C_1a_n/c_n^2 \) or \( C_2c_n^2 \). 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 (EC.18).) Expressions for the constants \( C_1 \) and \( C_2 \) are identified explicitly as part of this analysis.

**Remark 4 (Truncated KW Algorithm).** Theorem 1 requires the assumptions (F1) and (F2) to hold globally, which can be quite restrictive. This issue is also addressed in Kiefer and Wolfowitz (1952) 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 Andrásfai (1995) and Nemirovski et al. (2009) and references therein for some examples.

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

\[
X_{n+1} = \Pi_{I_{n+1}} \left( X_n + a_n \left( \frac{\hat{f}(X_n + c_n) - \hat{f}(X_n - c_n)}{c_n} \right) \right),
\]

(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.

**Remark 5 (Error Bounds for the Maximum).** Using a simple Taylor expansion and assumption (F1), we can derive from Theorem 1 upper bounds on \( f(x^*) - \mathbb{E}(X_n) \). Specifically, we have

\[
f(x^*) - \mathbb{E}(X_n) = |x^* - X_n| \cdot |f'(\xi_n)| \quad \text{for some } \xi_n \in (x^*, X_n)
\]

\[
\leq K_1|x^* - X_n| \cdot |\xi_n - x^*| \leq K_1(X_n - x^*)^2,
\]

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

\[
f(x^*) - \mathbb{E}(X_n) \leq \begin{cases} 
K_1C_1a_n/c_n^2 & \text{if } c_n^2 \leq \tau_1a_n \\
K_1C_2c_n^2 & \text{if } c_n^2 \geq \tau_2a_n,
\end{cases}
\]

(6)

where \( C_1, C_2, \tau_1, \tau_2 \) are defined in Theorem 1.

**Remark 6 (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 (1954b), with some obvious modifications to assumptions (F1) and (F2); see Theorem EC.2 in the electronic companion to this paper. The proof technique can also be applied to “random direction”-type algorithms such as SPSA, introduced by Spall (1992), and related variants (cf. Chen et al. 1999) by simply exploiting the tower property of conditional expectations; this is illustrated in Theorem EC.3 of the electronic companion.

**Remark 7 (Extensions to Root-Finding Problems).** Consider the setting described by Robbins and Monro (1951). The problem is to sequentially find the unique root \( x^* \) of \( g(x) = \xi \) using \( \hat{g}(\cdot) \) that are noisy observations of \( g(\cdot) \). Robbins and Monro (1951) consider the following stochastic approximation scheme:

\[
X_{n+1} = X_n + a_n(\xi - \hat{g}(X_n)) \quad n = 1, 2, \ldots
\]

(7)

Here, \( \hat{g}(X_n) \) is drawn according to the conditional distribution function \( H(y | X_n) \) with \( g(x) = \int y dH(y | x) \). The function \( g(x) \) is assumed to satisfy \((x - x^*)g(x) \geq K_0(x - x^*)^2\) and \( \mathbb{E}(g(x)^2) \leq K_1(1 + (x - x^*)^2) \) for all \( x \in \mathbb{R} \) and for some finite positive constants \( K_0, K_1 \). These are the standard assumptions in the RM context, cf. Benveniste et al. (1990). 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

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

(8)

for some finite positive constant \( C \) that can be explicitly identified. The proof follows almost verbatim the proof in Theorem 1. As a straightforward corollary of result (8), we conclude that the assumption \( \sum_{n=1}^{\infty} a_n^2 < \infty \) imposed in the majority of the stochastic approximation root-finding literature, is not required to prove convergence of the MSE to zero. Section EC.3.1 in the electronic companion extends this result to the multidimensional RM algorithm and contains the full statement of the theorem along with its proof.
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 nonincreasing sequence. Then the minimal order of the upper bound in (3) 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 8 (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 Dupac (1957) and Tsybakov and Polyak (1990). 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 9 (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. Nonnecessity of (KW3).
We exhibit sequences \( \{a_n\} \) and \( \{c_n\} \) that violate assumption (KW3) yet satisfy all assumptions of Theorem 1 and hence yield convergence of the mean-squared error to zero under the standard assumptions of (F1) and (F2). As mentioned in Remark 7, the nonnecessity of (KW3) in the context of sequentially estimating the point of maximum translates into non-necessity of the assumption \( \sum_{n=1}^{\infty} a_n^2 < \infty \) in the context of sequential root finding.

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 because \( c_n^4 \leq \tau^2 a_n \) with \( \tau = 1, \) 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, it is evident that \( \sum_{n=1}^{\infty} a_n^2/c_n^2 \) diverges, hence violating (KW3).

**Figure 1.** Illustration of the nonnecessity of (KW3).

Note. 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.

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 50,000 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(MSE) vs. log(log(n)) plot in Figure 1 is for iterations 5,000 to 10^6 and is \(-1.01 \pm 0.91 \) (95% confidence interval \((-1.11, -0.91))\), consistent with Theorem 1, which for \( a_n = 1/n \) and \( c_n = \sqrt{\log(n)/n} \) predicts a convergence rate of \( a_n/c_n^2 = 1/\log(n) \).

This choice of sequence only guarantees convergence of MSE and not almost sure convergence. Chen et al. (1999) develop necessary conditions for almost sure convergence of the iterates, and this choice of sequence violates those conditions.

2.3. The Special Case of “Quadratic-Like” Functions

The paper by Derman (1956) 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 \( O(1/n^{1+c}) \) for some \( c > 0 \). Next we revisit this analysis under the general framework developed in §2. We first restate the assumption given in Derman (1956).

(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)
The regression coefficient for this example is \( n \) up to \( 10^{-7} \), observe MSE convergence rate of order \( 1 \). By Proposition 2, for any quadratic function we should choose \( \tilde{\theta} > a \), to improve on the results of Derman (1956), eliminating the “for some \( \varepsilon > 0 \)” in his convergence result.

 Unlike Proposition 1, here the finite-difference approximation of the gradient matches the true gradient for any value of \( c_n \) due to the “quadratic-like” function structure. As a result, the trade-off between the two tuning sequences that determines the convergence rate in Theorem 1 does not exist in the setting of Proposition 3. Therefore, the optimal MSE convergence rate is achieved by setting the \( \{c_n\} \) sequence to a constant value, in violation of condition (KW1). As a side note, by not relying on Chung’s lemma we allow for more general sequences and use that to improve on the results of Derman (1956), 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 and independent standard normal noise at each function evaluation, Figure 2 contains a log-log plot of MSE versus iteration number \( n \) up to \( n = 10^8 \) steps in the algorithm. The MSE values are calculated using 1,000 independent runs of the algorithm. The regression coefficient for this example is \( -0.99 \) (95% confidence interval \((-1.02, -0.97)) \), which is close to the theoretical value of \(-1 \) predicted by Proposition 3.

**Figure 2.** Illustration of nonnecessity of (KW1) for “quadratic-like” functions.

Note. The log-log plot of MSE versus iteration number \( n \) shows that 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 (1957) derives the optimal rate of convergence for the basic KW algorithm (1) when the underlying function is thrice-differentiable. The result in Dupac (1957) is restricted to polynomial sequences as it again relies on the lemma by Chung (1954). We now revisit this problem and derive an analogue to Theorem 1. We restrict our attention to functions that satisfy (F1), (F2), and

\[
\begin{align*}
& (F4) \quad f'''(x) \text{ exists for all } x \in \mathbb{R} \text{ and } |f'''(x)| \leq T \text{ for some } T \in \mathbb{R}.
\end{align*}
\]

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

\[
\begin{align*}
& (S2') \quad c_n^0 \leq c_{n+1}^0 \frac{1 + Aa_{n+1}}{Aa_n} \text{ for all } n \geq 1, \\
& (S4') \quad \text{Either (i) } c_n^0/\alpha_n \leq \tau_1, \text{ or (ii) } c_n^0/\alpha_n \geq \tau_2, \text{ for all } n \geq 1.
\end{align*}
\]

**Remark 10.** Because 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 the 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 KW 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),

\[
\begin{align*}
\mathbb{E}(X_{n+1} - x^*)^2 & \leq \begin{cases} \\
C_1(a_n/c_n^2) & \text{if } c_n^0/\alpha_n \leq \tau_1, \\
C_2(c_n^2/a_n) & \text{if } c_n^0/\alpha_n \geq \tau_2,
\end{cases} \\
& \text{for all } n \geq 1 \text{ and for some finite positive constants } C_1 \text{ and } C_2.
\end{align*}
\]

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 \( O(a^{2/3}_{n+1}) \). 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. The proof follows the same steps as the proof of Proposition 1, and hence we omit the details.

**Proposition 4.** Let the assumption of Theorem 2 hold and suppose \( \{a_n\} \) is a nonincreasing sequence. Then the minimal order of the upper bound in (10) 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.

**Figure 3.** Oscillatory behavior of the truncated 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.,

\[
T = \sup\{n \geq 2: (X_n = u - c_n \text{ and } X_{n-1} = l + c_{n-1}) \text{ or } (X_n = l + c_n \text{ and } X_{n-1} = u - c_{n-1})\}, \tag{11}
\]

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., \( Y_t = f(x) + e_t \), with \( e_t \sim \mathcal{N}(0, \sigma^2) \) and \( \sigma = 1 \)) and \( X_1 = 30 \). The tuning sequences are \( a_n = 1/n \) and \( c_n = 1/\sqrt{n} \) as prescribed in §2. The oscillatory behavior can be observed for the first \( T = 9,960 \) 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 \( T = 9,960 \) 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 1,000 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 calculate the minimum positive integer \( \beta \) so that using \( a_{n+\beta} \) ensures that the function evaluations are within the interval; i.e., both \( X_n + c_n \in [I, u] \). 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 Figures 3(b) and 3(d) gives the relative frequency chart for \( X_{10,000} \) using 1,000 simulation replications.

**Remark 11 (Intuition for Shifting).** The idea of shifting the \( \{a_n\} \) sequence is inspired by close examination of the constants present in the upper bounds developed in §2.

For instance, if we seek to minimize the constant \( C \) in the upper bound for “quadratic-like” functions (see (EC.33) in the electronic companion), it is seen that this is achieved by balancing two terms. The first decreases with a decrease in the \( \{a_n\} \) sequence for large values of \( K_1 \). The second term increases as \( \{a_n\} \) decreases, so there is an evident trade-off.

The key observation is that when the gradient is steep, the first term dominates the second one, and therefore a smaller \( \{a_n\} \) sequence decreases the value of the constant \( C \) in our bound. Decreasing the step-size sequence \( \{a_n\} \) by a shift preserves more “energy” for future iterations, because it does not dampen the entire subsequent entries in \( \{a_n\} \) by the same multiplicative factor.

### 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 Nemirovski et al. (2009) for further discussion. If the tuning sequences do not satisfy this assumption, for instance if the multiplicative constant \( \theta_u \) in \( a_n = \theta_u/n \) is not large enough, a degraded convergence rate might result. As a simple example, similar to the one worked out in Nemirovski et al. (2009), consider \( f(x) = -0.001x^2 \) with \( a_n = 1/n \) and \( c_n = 1/\sqrt{n} \), 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

\[
X_n = 30 \prod_{m=1}^{n-1} \left(1 - \frac{1}{250m} \right) \geq \exp \left(- \sum_{m=1}^{n-1} \frac{1}{250m - 1} \right) \geq \frac{27}{n^{0.008}},
\]

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 (12). The relative frequency of \( X_{10,000} \) given in Figure 5(b) shows all sample paths exhibit a similar lack of convergence.

The problem of degraded convergence rate due to the constant \( \theta_u \) in \( a_n = \theta_u/n \) being too small relative to the magnitude of the gradient is present both in the RM and the KW algorithms. To tackle this problem in the RM framework, Ruppert (1988) and Polyak (1990) introduced the idea of averaging the iterates. They choose the \( \{a_n\} \) sequence to converge to zero slower than \( 1/n \) and define \( \bar{X}_n = (\sum_{i=1}^n X_i)/n \), where \( \{X_i\} \) is the sequence generated by the RM algorithm with this choice of \( \{a_n\} \) sequence. They
Figure 5. Degraded convergence rate due to a small step size.

Notes. Panel (a) shows a sample path in the KW algorithm for \( f(x) = -0.001x^2 \) with \( a_n = 1/n, c_n = 1/\sqrt{n} \), and \( \sigma = 0.001 \). The \( \lambda < 4K_0 \) assumption in (S1) and (S2) is violated. From Table 3, the convergence rate of the MSE is \(-0.008 \pm 1.9 \times 10^{-6}\). Panel (b) is the relative frequency chart for \( \bar{X}_{10,000} \) exhibiting poor performance in all 1,000 simulated sample paths. Panel (c) shows a sample path of Polyak-Ruppert averages of iterates generated in the exact same setting, but with using \( a_n = \log(n)/n \). The MSE of the averages converges at a rate estimated to be \(-0.05 \pm 1.6 \times 10^{-5}\). Relative frequency chart for \( \bar{X}_{10,000} \) given in Panel (d) shows the poor convergence of the averages is present in all 1,000 sample paths. Panel (e) shows a sample path in the scaled-and-shifted KW algorithm in the same setting using the same noise random sequence. After four scale-ups, the \( \{a_n\} \) sequence becomes \( a_n = 1.987/n \) and shifting is not needed. From Table 3, the scaling results in an MSE convergence rate estimate of \(-0.53 \pm 0.05\), which recovers the optimal rate. As seen in Panel (f), this is observed in all of the 1,000 simulated sample paths.

prove that with these changes, \( n^{1/2}(\bar{X}_n - x^*) \) converges in distribution to a normally distributed random variable with zero mean and variance that is independent of the constant in the \( \{a_n\} \) sequence. Thus, the method achieves the optimal convergence rate for the RM framework independent of the choice of the constant in the tuning sequence. A corresponding result is developed by Dippon and Renz (1997) for the KW algorithm. In particular, for twice differentiable functions, the choice of \( a_n = \theta_n \log(n)/n \) combined with iterate averaging guarantees the optimal convergence rate in the KW framework. This class of algorithms serve as a natural benchmark for our proposed algorithm.

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 maximum number of forced boundary hits is a user-specified parameter set to four in all our numerical experiments. Figure 5(e) 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(f), the scaled-and-shifted KW algorithm improves the convergence on all 1,000 simulated samples, although unlike the Polyak-Ruppert scheme, there are no theoretical guarantees for the SSKW algorithm.

The performance of Polyak-Ruppert averaging, with $a_n = \log(n+1)/n$, is displayed in Figure 5(c) under the same setting as in panel (a) and using identical random numbers. A slight improvement is noted relative to the TKW results given in panels (a) and (b), but the observed convergence behavior of Polyak-Ruppert averaging in this example is quite poor. In particular, the MSE exponent is calculated to be $-0.05$ (see Table 3 for a confidence interval), which is far from the guaranteed asymptotically optimal rate of $-0.5$. Figure 5(d) contains the relative frequency of final estimates and shows that all 1,000 sample paths exhibit similar poor performance.

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 might give rise to excessive noise in the gradient estimates. As a consequence, even at the boundaries of the truncation interval, the algorithm might 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 $f(x) = 1,000 \cos(\pi x/100)$ 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., $Y_i = f(x) + \epsilon_i$, with $\epsilon_i \sim N(0, \sigma^2)$ and $X_1 = 30$. The main difference is that we assume a large noise level given by $\sigma = 1,000$. 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 $X_{1,000}$ 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 $\gamma > 1$ 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 $x^*$). This situation is one where the error in the

Figure 6. Noisy gradient estimate problem.
gradient estimates is dominated by the noise term, because by assumption (F2) the true gradient at the boundary has to point toward $x^*$. We also make sure that the scaled-up $\{c_n\}$ sequence does not exceed an upper bound $c_{\text{max}}$, 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 $\{a_i\}$ 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 $f(x) = 1,000 \cos(\pi x/100)$ with the same parameters and random numbers. The $\{c_n\}$ sequence is scaled up four times at early stages of the algorithm, while the $\{a_i\}$ sequence is neither shifted nor scaled. With this adaptive tuning of the sequences, the iterates move toward the point of maximum much faster, and this behavior is consistent throughout 1,000 sample paths as shown in Figure 6(d). In this setting, the scaled-and-shifted KW algorithm achieves an MSE convergence rate of $-0.53 \pm 0.02$ (see Table 5).

3.2. Numerical Results

In this section we provide numerical results for the scaled-and-shifted KW algorithm, as described in the appendix, which combines the remedies described previously. Results for the truncated KW algorithm are given for comparison. We also provide the results for Polyak-Ruppert averaging for the second example below, which illustrate the rate degradation problem (because that scheme is aimed only for the second example below, which illustrate the rate degradation problem (because that scheme is aimed only at mitigating this particular issue). Algorithm sample paths are generated for 10,000 iterations. The standard errors are within 7% of the MSE values in all cases. Empirical convergence rates are calculated using 1000 independent replications. Statistics on the adaptations to the sequences are given in Table 2.

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, which is calculated using 1,000 independent replications. Statistics on the adaptations to the sequences 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 $\{a_n = 1/n\}$ 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 estimated convergence rate of truncated KW algorithm is close to zero, i.e., it is not converging for all practical purposes. The Polyak-Ruppert averaging idea improves on this slightly.

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>100</th>
<th>1,000</th>
<th>10,000</th>
<th>5%</th>
<th>Median</th>
<th>95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>8.7</td>
<td>0.6</td>
<td>0.08</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>[2.469]</td>
<td>[2.482]</td>
<td>[16]</td>
<td>[9,960]</td>
<td>[9,960]</td>
<td>[9,960]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8.5</td>
<td>0.6</td>
<td>0.08</td>
<td>27</td>
<td>27</td>
<td>29</td>
</tr>
<tr>
<td>[2.469]</td>
<td>[2.482]</td>
<td>[15]</td>
<td>[9,959]</td>
<td>[9,960]</td>
<td>[9,960]</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7.2</td>
<td>0.6</td>
<td>0.2</td>
<td>22</td>
<td>27</td>
<td>31</td>
</tr>
<tr>
<td>[2.469]</td>
<td>[2.482]</td>
<td>[12]</td>
<td>[9,957]</td>
<td>[9,959]</td>
<td>[9,961]</td>
<td></td>
</tr>
</tbody>
</table>

Notes. This table shows the MSE calculated at iterations 100, 1,000, and 10,000; the convergence rate estimate for the MSE and the 5th and 95th percentiles; 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.
but the estimated convergence rate is still quite far from the optimal rate of $-0.5$. On the other hand, SSKW algorithm significantly improves the convergence behavior, recovering the optimal rate. Table 4 presents statistics on the adaptations to the sequences and shows that there is significant scaling up in the $\{a_n\}$ sequence, as well as the total scale-up factor $\beta$, 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 ($\sigma$).

**Example 3.** The last test function is \( f(x) = 1,000 \cos(\pi x/10); \) 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, because the assumption $A < 4K_0$ is satisfied for the initial choice of the $\{a_n\}$ sequence, the scaling up of the $\{c_n\}$ sequence decreases performance in terms of MSE. But because 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. All the numbers in Tables 3 and 4 are calculated using 2,000 independent replications.

In all three examples, scaling and shifting the tuning sequences resulted in vastly improved finite-time performance, and essentially optimal estimates of the rate of convergence (for example, the improvement in the MSE can be as high as a factor of 150,000). In instances where the original choice of the sequences is a good fit to the characteristics of the underlying function, and where the TKW algorithm does seem to converge at the optimal rate, scaling and shifting does not degrade the convergence rate. In these cases, the scaled-and-shifted KW algorithm outperforms its adaptive counterpart in terms of MSE.

### Table 2. Tuning sequence statistics for \( f(x) = -x^4 \).

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>SSKW</td>
<td>0.05</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>0.05</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>TKW</td>
<td>0.05</td>
<td>0.02</td>
</tr>
<tr>
<td>0.01</td>
<td>SSKW</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>TKW</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td>0.1</td>
<td>SSKW</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>TKW</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td>1</td>
<td>SSKW</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>TKW</td>
<td>0.1</td>
<td>0.01</td>
</tr>
</tbody>
</table>

**Note.** The 5th and 95th percentiles along with the median values are given for the total scale-up factor $\alpha$, and the total shift amount $\beta$ in the $\{a_n\}$ tuning sequence, as well as the total 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 ($\sigma$).

### Table 3. Comparison of the scaled-and-shifted KW algorithm, the truncated KW algorithm and the KW algorithm with Polyak-Ruppert (PR) averaging for \( f(x) = -0.001x^2 \).

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>Alg.</th>
<th>MSE</th>
<th>Convergence</th>
<th>Length of oscillatory period</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>1,000</td>
<td>10,000</td>
</tr>
<tr>
<td>0.001</td>
<td>SSKW</td>
<td>0.05</td>
<td>0.02</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>843</td>
<td>773</td>
<td>679</td>
</tr>
<tr>
<td></td>
<td>TKW</td>
<td>863</td>
<td>848</td>
<td>833</td>
</tr>
<tr>
<td>0.01</td>
<td>SSKW</td>
<td>5.1</td>
<td>1.7</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>843</td>
<td>773</td>
<td>679</td>
</tr>
<tr>
<td></td>
<td>TKW</td>
<td>863</td>
<td>848</td>
<td>832</td>
</tr>
<tr>
<td>0.1</td>
<td>SSKW</td>
<td>179</td>
<td>58</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>843</td>
<td>772</td>
<td>678</td>
</tr>
<tr>
<td></td>
<td>TKW</td>
<td>863</td>
<td>847</td>
<td>832</td>
</tr>
<tr>
<td>1</td>
<td>SSKW</td>
<td>243</td>
<td>73</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>844</td>
<td>784</td>
<td>696</td>
</tr>
<tr>
<td></td>
<td>TKW</td>
<td>863</td>
<td>848</td>
<td>833</td>
</tr>
</tbody>
</table>

**Note.** This table shows the MSE calculated at iterations 100, 1,000, and 10,000; the convergence rate for the MSE, and the 5th and 95th percentiles, and the median of the length of the oscillatory periods at different noise levels, $\sigma$.
Table 5. Comparison of the scaled-and-shifted KW algorithm and the truncated KW algorithm for $f(x) = 1,000 \cos(\pi x / 100)$.

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>MSE</th>
<th>Convergence</th>
<th>Length of oscillatory period</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>100, 1000, 10,000</td>
<td>Rate 5% Median 95%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>[6] [1.9] [0.6]</td>
<td>$[-0.50 \pm 0.03]$ 4 4 6</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>[530] [207] [61]</td>
<td>$[-0.53 \pm 0.03]$ 0 [0] [0]</td>
</tr>
</tbody>
</table>

Note. This table shows the MSE calculated at iterations 100, 1,000, and 10,000; the convergence rate for the MSE, and the 5th and 95th percentiles 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.

Table 6. Modifications in the tuning sequences for $f(x) = 1,000 \cos(\pi x / 100)$.

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>5% Median 95%</th>
<th>$\alpha$</th>
<th>5% Median 95%</th>
<th>$\beta$</th>
<th>5% Median 95%</th>
<th>$\gamma$</th>
<th>5% Median 95%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>4.5 6.9</td>
<td>15.1</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1.8 7.3</td>
<td>33</td>
<td>0</td>
<td>4</td>
<td>37</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1,000</td>
<td>1.0 1.8</td>
<td>12</td>
<td>0</td>
<td>6</td>
<td>42</td>
<td>16</td>
</tr>
</tbody>
</table>

Note. The 5th and 95th percentiles along with the median values are given for the total scale-up factor, $\alpha$ and the total shift amount, $\beta$ in (a.μ) sequence as well as the total scale-up factor for the (c.μ) sequence, $\gamma$, for various noise levels ($\sigma$).

instances (such as example 3 with $\sigma = 10$ given in Table 5), the MSE values for the SSK algorithm are slightly worse than those of the TKW algorithm mainly because of the scale-ups in the (a.μ) sequence, which forces boundary hits.

4. Electronic Companion
An electronic companion to this paper is available as part of the online version that can be found at http://or.journal.informs.org/.

Endnotes
1. The proof of Theorem 1 for the truncated KW algorithm is identical until equation (EC.9), where we now have

$$Z_{n+1} := (X_{n+1} - x)^2$$

$$= \left[ \Pi_{h_{a-1}} (X_n + a_n \left( \hat{f}(X_n + c_n) - \hat{f}(X_n - c_n) \right) / c_n ) - x \right]^2$$

$$\leq \left[ X_n + a_n \left( \hat{f}(X_n + c_n) - \hat{f}(X_n - c_n) \right) / c_n ) - x \right]^2,$$

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 $W_n := X_n + a_n (\hat{f}(X_n + c_n) - \hat{f}(X_n - c_n) / c_n )$. The remainder of the proof is identical.

2. If the starting point of optimization is known to be close to the optimal solution, then this remedy might remove the value of a good starting point. However, if the rate degradation problem is known not to be an issue, the user can set the number of forced boundary hits, parameter $h_0$, to zero. Also, if it’s known a priori that the optimal solution is close to the starting point, one can set tighter bounds on the truncation interval.

Acknowledgments
This work was supported by Credit Suisse and NSF grants DMI-0447652 and DMS-0914539.

References


