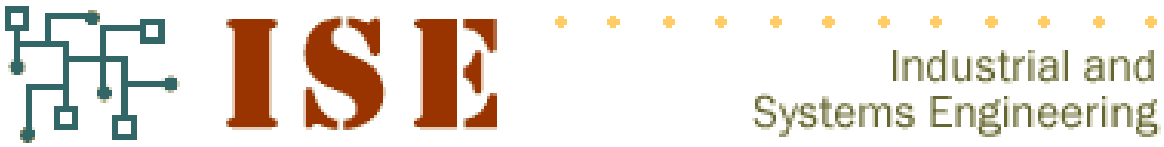

R-Linear Convergence of Limited Memory Steepest Descent

Frank E. Curtis and Wei Guo

Department of Industrial and Systems Engineering, Lehigh University, USA

COR@L Technical Report 16T-010

R-Linear Convergence of Limited Memory Steepest Descent*

Frank E. Curtis[†1] and Wei Guo[‡1]

[1]Department of Industrial and Systems Engineering, Lehigh University, USA

October 12, 2016

Abstract

The limited memory steepest descent method (LMSD) proposed by Fletcher is an extension of the Barzilai-Borwein "two-point step size" strategy for steepest descent methods for solving unconstrained optimization problems. It is known that the Barzilai-Borwein strategy yields a method with an R-linear rate of convergence when it is employed to minimize a strongly convex quadratic. This paper extends this analysis for LMSD, also for strongly convex quadratics. In particular, it is shown that the method is R-linearly convergent for any choice of the history length parameter. The results of numerical experiments are provided to illustrate behaviors of the method that are revealed through the theoretical analysis.
Keywords: unconstrained optimization; steepest descent methods; Barzilai-Borwein methods; limited memory methods; quadratic optimization; R-linear rate of convergence

Dedicated to Roger Fletcher and Jonathan Borwein whose contributions continue to inspire many in the fields of nonlinear optimization and applied mathematics

1 Introduction

For solving unconstrained nonlinear optimization problems, one of the simplest and most widely used techniques is *steepest descent* (SD). This refers to any strategy in which, from any solution estimate, a productive step is obtained by moving some distance along the negative gradient of the objective function, i.e., the direction along which function descent is steepest.

While SD methods have been studied for over a century and employed in numerical software for decades, a unique and powerful instance came about relatively recently in the work by [1], where a "two-point step size" strategy is proposed and analyzed. The resulting SD method, commonly referred to as the BB method, represents an effective alternative to other SD methods that employ an exact or inexact line search when computing the stepsize in each iteration.

The theoretical properties of the BB method are now well-known when it is employed to minimize an n-dimensional strongly convex quadratic objective function. Such objective functions are interesting in their own right, but one can argue that such analyses also characterize the behavior of the method in the neighborhood of a strong local minimizer of any smooth objective function. In the original work (i.e., [1]), it is shown that the method converges R-superlinearly when $n = 2$. In [5], it is shown that the method converges from any starting point for any natural number n, and in [2] it is shown that the method converges R-linearly for any such n.

*This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics, Early Career Research Program under contract number DE–SC0010615, as well as by the U.S. National Science Foundation under Grant Nos. DMS–1319356 and CCF-1618717.

[†]E-mail: frank.e.curtis@gmail.com

[‡]E-mail: weg411@lehigh.edu

In each iteration of the BB method, the stepsize is determined by a computation involving the displacement in the gradient of the objective observed between the current iterate and the previous iterate. As shown in [3], this idea can be extended to a *limited memory steepest descent* (LMSD) method in which a *sequence* of m stepsizes is computed using the displacements in the gradient over the previous m steps. This extension can be motivated by the observation that these displacements lie in a Krylov subspace determined by a gradient previously computed in the algorithm, which in turn yields a computationally efficient strategy for computing m distinct eigenvalue estimates of the Hessian (i.e., matrix of second derivatives) of the objective function. The reciprocals of these eigenvalue estimates represent reasonable stepsize choices. Indeed, if the eigenvalues of the Hessian are computed exactly, then the algorithm terminates in a finite number of iterations; see [3] and §2.

In [3], it is shown that the proposed LMSD method converges from any starting point when it is employed to minimize a strongly convex quadratic function. However, to the best of our knowledge, the convergence rate of the method for $m > 1$ has not yet been analyzed. The main purpose of this paper is to show that this LMSD method converges R-linearly when employed to minimize such a function. Our analysis builds upon the analyses in [3] and [2].

We mention at the outset that numerical evidence has shown that the practical performance of the BB method is typically much better than known convergence proofs suggest; in particular, the empirical rate of convergence is often Q-linear with a contraction constant that is better than that observed for a basic SD method. Based on such evidence, we do not claim that the convergence results proved in this paper fully capture the practical behavior of LMSD methods. To explore this claim, we present the results of numerical experiments that illustrate our convergence theory and demonstrate that the practical performance of LMSD can be even better than the theory suggests. We conclude with a discussion of possible explanations of why this is the case for LMSD, in particular by referencing a known finite termination result for a special (computationally expensive) variant of the algorithm.

Organization In §2, we formally state the problem of interest, notation to be used throughout the paper, Fletcher's LMSD algorithm, and a finite termination property for it. In §3, we prove that the LMSD algorithm is R-linearly convergent for any history length. The theoretical results proved in §3 are demonstrated numerically in §4 and concluding remarks are presented in §5.

Notation The set of real numbers (i.e., scalars) is denoted as $\mathbb{R}$, the set of nonnegative real numbers is denoted as $\mathbb{R}_+$, the set of positive real numbers is denoted as $\mathbb{R}_{++}$, and the set of natural numbers is denoted as $\mathbb{N} := \{1, 2, \dots\}$. A natural number as a superscript is used to denote the vector-valued extension of any of these sets—e.g., the set of n-dimensional real vectors is denoted as $\mathbb{R}^n$—and a Cartesian product of natural numbers as a superscript is used to denote the matrix-valued extension of any of these sets—e.g., the set of $n \times n$ real matrices is denoted as $\mathbb{R}^{n\times n}$. A finite sequence of consecutive positive integers of the form $\{1, \dots, n\} \subset \mathbb{N}$ is denoted using the shorthand $[n]$. Subscripts are used to refer to a specific element of a sequence of quantities, either fixed or generated by an algorithm. For any vector $v \in \mathbb{R}^n$, its Euclidean (i.e., ℓ_2) norm is denoted by $\|v\|$.

2 Fundamentals

In this section, we state the optimization problem of interest along with corresponding definitions and concepts to which we will refer throughout the remainder of the paper. We then state Fletcher's LMSD algorithm and prove a finite termination property for it similar to that proved in [3].

2.1 Problem Statement

Consider the problem to minimize a strongly convex quadratic function $f : \mathbb{R}^n \to \mathbb{R}$ defined by a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$ and vector $b \in \mathbb{R}^n$, namely,

$$\min_{x\in\mathbb{R}^n} \; f(x), \quad \text{where} \quad f(x) = \tfrac{1}{2}x^T Ax - b^T x. \tag{1}$$

Formally, we make the following assumption about the problem data.

Assumption 2.1. *The matrix A in problem (1) has $r \leq n$ distinct eigenvalues denoted by*

$$\lambda_{(r)} > \cdots > \lambda_{(1)} > 0. \tag{2}$$

Consequently, this matrix yields the eigendecomposition $A = Q\Lambda Q^T$, where

$$\begin{aligned} Q = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix} \quad & \textit{is orthogonal} \\ \textit{and} \;\; \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) \quad & \textit{with} \;\; \lambda_n > \cdots > \lambda_1 > 0 \\ & \textit{and} \;\; \lambda_i \in \{\lambda_{(1)}, \ldots, \lambda_{(r)}\} \;\; \textit{for all} \;\; i \in [n]. \end{aligned} \tag{3}$$

The eigendecomposition of A defined in Assumption 2.1 plays a crucial role in our analysis. In particular, we will make extensive use of the fact that any gradient of the objective function computed in the algorithm, a vector in $\mathbb{R}^n$, can be written as a linear combination of the columns of the orthogonal matrix Q. This will allow us to analyze the behavior of the algorithm componentwise according to the weights in these linear combinations corresponding to the sequence of computed objective gradients. Such a strategy has been employed in all of the aforementioned articles on BB and LMSD.

2.2 Limited memory steepest descent (LMSD) method

Fletcher's limited memory steepest descent method is stated as Algorithm LMSD. The iterate update in the algorithm is the standard update in an SD method: each subsequent iterate is obtained from the current iterate minus a multiple of the gradient of the objective function evaluated at the current iterate. With this update at its core, Algorithm LMSD operates in cycles. At $x_{k,1} \in \mathbb{R}^n$ representing the initial point of the kth cycle, a sequence of m positive stepsizes $\{\alpha_{k,j}\}_{j\in[m]}$ are selected to be employed in an inner cycle composed of m updates, the result of which is set as the initial point for cycle $k+1$.

Once such an inner cycle has been performed, the stepsizes to be employed in the next cycle are computed as the reciprocals of estimates of eigenvalues of A. [3] describes how these can be obtained in one of three ways, all offering the same estimates (in exact arithmetic). The most intuitive definition is that, for cycle $k+1$, the estimates come as the eigenvalues of $T_k := Q_k^T A Q_k$, where $Q_k \in \mathbb{R}^{n\times m}$ satisfying $Q_k^T Q_k = I$ is defined by a thin QR factorization of the matrix of kth cycle gradients, i.e., for some upper triangular matrix $R_k \in \mathbb{R}^{m\times m}$, such a factorization satisfies the equation

$$Q_k R_k = G_k := \begin{bmatrix} g_{k,1} & \cdots & g_{k,m} \end{bmatrix}. \tag{4}$$

(For now, let us assume that G_k has linearly independent columns, in which case the matrix R_k in (4) is nonsingular. For a discussion of situations when this is not the case, see Remark 2.2 later on.) Practically, however, obtaining T_k in this manner requires multiplications with A as well as the storage of the n-vectors composing the columns of Q_k. Both can be avoided by obtaining the gradient at the initial point of cycle $k+1$, namely $g_{k+1,1} \equiv g_{k,m+1}$, as well as the matrix of kth-cycle reciprocal stepsizes

$$J_k \leftarrow \begin{bmatrix} \alpha_{k,1}^{-1} & & \\ -\alpha_{k,1}^{-1} & \ddots & \\ & \ddots & \alpha_{k,m}^{-1} \\ & & -\alpha_{k,m}^{-1} \end{bmatrix}, \tag{5}$$

computing (upper triangular) R_k and r_k from the partially extended Cholesky factorization

$$G_k^T \begin{bmatrix} G_k & g_{k,m+1} \end{bmatrix} = R_k^T \begin{bmatrix} R_k & r_k \end{bmatrix}, \tag{6}$$

then computing

$$T_k \leftarrow \begin{bmatrix} R_k & r_k \end{bmatrix} J_k R_k^{-1}. \tag{7}$$

Fletcher's third approach, which also avoids multiplications with A, is to compute

$$T_k \leftarrow \begin{bmatrix} R_k & Q_k^T g_{k,m+1} \end{bmatrix} J_k R_k^{-1}. \tag{8}$$

However, this is less efficient than using (7) due to the need to store Q_k and since the QR factorization of G_k requires $\sim m^2 n$ flops, as opposed to the $\sim \frac{1}{2} m^2 n$ flops required for (7); see [3].

Algorithm LMSD Limited Memory Steepest Descent Method

choose an initial point $x_{1,1} \in \mathbb{R}^n$, history length $m \in \mathbb{N}$, and termination tolerance $\epsilon \in \mathbb{R}_+$
choose stepsizes $\{\alpha_{1,j}\}_{j \in [m]} \subset \mathbb{R}_{++}$
compute $g_{1,1} \leftarrow \nabla f(x_{1,1})$
if $\|g_{1,1}\| \leq \epsilon$, **then return** $x_{1,1}$
for $k \in \mathbb{N}$ **do**
    for $j \in [m]$ **do**
        set $x_{k,j+1} \leftarrow x_{k,j} - \alpha_{k,j} g_{k,j}$
        compute $g_{k,j+1} \leftarrow \nabla f(x_{k,j+1})$
        if $\|g_{k,j+1}\| \leq \epsilon$, **then return** $x_{k,j+1}$
    end for
    set $x_{k+1,1} \leftarrow x_{k,m+1}$ and $g_{k+1,1} \leftarrow g_{k,m+1}$
    set G_k by (4) and J_k by (5)
    compute R_k and r_k to satisfy (6) and set T_k by (7)
    set $\{\theta_{k,j}\}_{j \in [m]} \subset \mathbb{R}_{++}$ as the eigenvalues of T_k in decreasing order
    set $\{\alpha_{k+1,j}\}_{j \in [m]} \leftarrow \{\theta_{k,j}^{-1}\}_{j \in [m]} \subset \mathbb{R}_{++}$
end for

The choice to order the eigenvalues of T_k in decreasing order is motivated by [3]. In short, this ensures that the stepsizes in cycle $k+1$ are ordered from smallest to largest, which improves the likelihood that the objective function and the norm of the objective gradient decrease monotonically, at least initially, in each cycle. This ordering is not essential for our analysis, but is a good choice for any implementation of the algorithm; hence, we state the algorithm to employ this ordering.

One detail that remains for a practical implementation of the method is how to choose the initial stepsizes $\{\alpha_{1,j}\}_{j \in [m]} \subset \mathbb{R}_{++}$. This choice has no effect on the theoretical results proved in this paper, though our analysis does confirm the fact that the practical performance of the method can improved if one has the knowledge to choose one or more stepsizes exactly equal to reciprocals of eigenvalues of A; see §2.3. Otherwise, one can either provide a full set of m stepsizes or carry out an initialization phase in which the first few cycles are shorter in length, dependent on the number of objective gradients that have been observed so far; see [3] for further discussion on this matter.

Remark 2.2. *In* (4)*, if G_k for some $k \in \mathbb{N}$ does not have linearly independent columns, then R_k is singular and the formulas* (7) *and* (8) *are invalid, meaning that the employed approach is not able to provide m eigenvalue estimates for cycle k. As suggested in [3], an implementation of the method can address this by iteratively removing "older" columns of G_k until the columns form a linearly independent set of vectors, in which case the approach would be able to provide $\tilde{m} \leq m$ stepsizes for the subsequent (shortened) cycle. We advocate such an approach in practice and, based on the results proved in this paper, conjecture that the convergence rate of the algorithm would be R-linear. However, the analysis for such a method would be extremely cumbersome given that the number of iterations in each cycle might vary from one cycle to the*

next within a single run of the algorithm. Hence, in our analysis in §3, we assume that G_k has linearly independent columns for all $k \in \mathbb{N}$. In fact, we go further and assume that $\|R_k^{-1}\|$ is bounded proportionally to the reciprocal of the norm of the objective gradient at the first iterate in cycle k (meaning that the upper bound diverges as the algorithm converges to the minimizer of the objective function). These norms are easily computed in an implementation of the algorithm; hence, we advocate that a procedure of iteratively removing "older" columns of G_k would be based on observed violations of such a bound. See the discussion following Assumption 3.4 in §3.

2.3 Finite Termination Property of LMSD

If, for some $k \in \mathbb{N}$ and $j \in [m]$, the stepsizes in Algorithm LMSD up through iteration $(k, j) \in \mathbb{N} \times [m]$ include the reciprocals of all of the $r \leq n$ distinct eigenvalues of A, then the algorithm terminates by the end of iteration (k, j) with $x_{k,j+1}$ yielding $\|g_{k,j+1}\| = 0$. This is shown in the following lemma and theorem, which together demonstrate and extend the arguments made in §2 of [3].

Lemma 2.3. *Under Assumption 2.1, for each $(k, j) \in \mathbb{N} \times [m]$, there exist weights $\{d_{k,j,i}\}_{i \in [n]}$ such that $g_{k,j}$ can be written as a linear combination of the columns of Q in (3), i.e.,*

$$g_{k,j} = \sum_{i=1}^{n} d_{k,j,i} q_i. \tag{9}$$

Moreover, these weights satisfy the recursive property

$$d_{k,j+1,i} = (1 - \alpha_{k,j}\lambda_i) d_{k,j,i} \quad \text{for all} \quad (k, j, i) \in \mathbb{N} \times [m] \times [n]. \tag{10}$$

Proof. Since $g_{k,j} = Ax_{k,j} - b$ for all $(k, j) \in \mathbb{N} \times [m]$, it follows that

$$\begin{aligned}
& x_{k,j+1} = x_{k,j} - \alpha_{k,j} g_{k,j}, \\
\implies \quad & Ax_{k,j+1} = Ax_{k,j} - \alpha_{k,j} Ag_{k,j}, \\
\implies \quad & g_{k,j+1} = g_{k,j} - \alpha_{k,j} Ag_{k,j}, \\
\implies \quad & g_{k,j+1} = (I - \alpha_{k,j} A) g_{k,j}, \\
\implies \quad & g_{k,j+1} = (I - \alpha_{k,j} Q \Lambda Q^T) g_{k,j},
\end{aligned}$$

from which one obtains that

$$\sum_{i=1}^{n} d_{k,j+1,i} q_i = \sum_{i=1}^{n} d_{k,j,i} (I - \alpha_{k,j} Q \Lambda Q^T) q_i = \sum_{i=1}^{n} d_{k,j,i} (q_i - \alpha_{k,j} \lambda_i q_i) = \sum_{i=1}^{n} d_{k,j,i} (1 - \alpha_{k,j} \lambda_i) q_i.$$

The result then follows since the columns of Q form an orthogonal basis of $\mathbb{R}^n$. □

Theorem 2.4. *Suppose that Assumption 2.1 holds and that Algorithm LMSD is run with termination tolerance $\epsilon = 0$. If, for some $(k, j) \in \mathbb{N} \times [m]$, the set of computed stepsizes up through iteration (k, j) includes all of the values $\{\lambda_{(l)}^{-1}\}_{l \in [r]}$, then, at the latest, the algorithm terminates finitely at the end of iteration (k, j) with $x_{k,j+1}$ yielding $\|g_{k,j+1}\| = 0$.*

Proof. Consider any $(k, j) \in \mathbb{N} \times [m]$ such that the stepsize is equal to the reciprocal of an eigenvalue of A, i.e., $\alpha_{k,j} = \lambda_{(l)}^{-1}$ for some $l \in [r]$. By Lemma 2.3, it follows that

$$d_{k,j+1,i} = (1 - \alpha_{k,j}\lambda_i) d_{k,j,i} = (1 - \lambda_{(l)}^{-1}\lambda_i) d_{k,j,i} = 0 \quad \text{for all} \quad i \in [n] \quad \text{such that} \quad \lambda_i = \lambda_{(l)}.$$

Along with the facts that Lemma 2.3 also implies

$$d_{k,j,i} = 0 \implies d_{k,j+1,i} = 0 \quad \text{for all} \quad (k, j) \in \mathbb{N} \times [m]$$

and $x_{k+1,1} \leftarrow x_{k,m+1}$ (and $g_{k+1,1} \leftarrow g_{k,m+1}$) for all $k \in \mathbb{N}$, the desired conclusion follows. □

Remark 2.5. *Theorem 2.4 implies that Algorithm LMSD will converge finitely by the end of the second cycle if $m \geq r$ and the eigenvalues of T_1 include all eigenvalues $\{\lambda_{(l)}\}_{l \in [r]}$. This is guaranteed, e.g., when the first cycle involves $m = n$ steps and G_1 has linearly independent columns.*

3 R-Linear Convergence Rate of LMSD

Our primary goal in this section is to prove that Algorithm LMSD converges R-linearly for any choice of the history length parameter $m \in \mathbb{N}$. For context, we begin by citing two known convergence results that apply for Algorithm LMSD, then turn our attention to our new convergence rate results.

3.1 Known Convergence Properties of LMSD

In the Appendix of [3], the following convergence result is proved for Algorithm LMSD. The theorem is stated slightly differently here only to account for our different notation.

Theorem 3.1. *Suppose that Assumption 2.1 holds and that Algorithm LMSD is run with termination tolerance $\epsilon = 0$. Then, either $g_{k,j} = 0$ for some $(k, j) \in \mathbb{N} \times [m]$ or the sequences $\{g_{k,j}\}_{k=1}^{\infty}$ for each $j \in [m]$ converge to zero.*

As a consequence of this result, we may conclude that if Algorithm LMSD does not terminate finitely, then, according to the relationship (9), the following limits hold:

$$\lim_{k\to\infty} g_{k,j} = 0 \quad \text{for each} \quad j \in [m] \quad \text{and} \tag{11a}$$

$$\lim_{k\to\infty} d_{k,j,i} = 0 \quad \text{for each} \quad (j, i) \in [m] \times [n]. \tag{11b}$$

Fletcher's result, however, does not illuminate the rate at which these sequences converge to zero. Only for the case of $m = 1$ in which Algorithm LMSD reduces to a BB method do the following results from [2] (see Lemma 2.4 and Theorem 2.5 therein) provide a convergence rate guarantee.

Lemma 3.2. *Suppose that Assumption 2.1 holds and that Algorithm LMSD is run with history length $m = 1$ and termination tolerance $\epsilon = 0$. Then, there exists $K \in \mathbb{N}$, dependent only on (λ_1, λ_n), such that*

$$\|g_{k+K,1}\| \leq \tfrac{1}{2}\|g_{k,1}\| \quad \text{for all} \quad k \in \mathbb{N}.$$

Theorem 3.3. *Suppose that Assumption 2.1 holds and that Algorithm LMSD is run with history length $m = 1$ and termination tolerance $\epsilon = 0$. Then, either $g_{k,1} = 0$ for some $k \in \mathbb{N}$ or*

$$\|g_{k,1}\| \leq c_1 c_2^k \|g_{1,1}\| \quad \text{for all} \quad k \in \mathbb{N},$$

where, with $K \in \mathbb{N}$ from Lemma 3.2, the constants are defined as

$$c_1 := 2\left(\frac{\lambda_n}{\lambda_1} - 1\right)^{K-1} \quad \text{and} \quad c_2 := 2^{-1/K} \in (0, 1).$$

Overall, the computed gradients vanish R-linearly with constants that depend only on (λ_1, λ_n).

3.2 R-Linear Convergence Rate of LMSD for Arbitrary $m \in \mathbb{N}$

Our goal in this subsection is to build upon the proofs of the results stated in the previous subsection (as given in the cited references) to show that Algorithm LMSD possesses an R-linear rate of convergence for any $m \in \mathbb{N}$. More precisely, our goal is to show that the gradients computed by the algorithm vanish R-linearly with constants that depend only on the spectrum of the data matrix A. Formally, for simplicity and brevity in our analysis, we make the following standing assumption throughout this section.

Assumption 3.4. *Assumption 2.1 holds, as do the following:*

(i) Algorithm LMSD is run with $\epsilon = 0$ and $g_{k,j} \neq 0$ for all $(k, j) \in \mathbb{N} \times [m]$.

(ii) For all $k \in \mathbb{N}$, the matrix G_k has linearly independent columns. Further, there exists a scalar $\rho \geq 1$ such that, for all $k \in \mathbb{N}$, the nonsingular matrix R_k satisfies $\|R_k^{-1}\| \leq \rho \|g_{k,1}\|^{-1}$.

Assumption 3.4(i) is reasonable since, in any situation in which the algorithm terminates finitely, all of our results hold for the iterations prior to that in which the algorithm terminates. Hence, by proving that the algorithm possesses an R-linear rate of convergence for cases when it does not terminate finitely, we claim that it possesses such a rate in all cases. As for Assumption 3.4(ii), first recall Remark 2.2. In addition, the bound on the norm of the inverse of R_k is reasonable since, in the case of $m = 1$, one finds that $Q_k R_k = G_k = g_{k,1}$ has $Q_k = g_{k,1}/\|g_{k,1}\|$ and $R_k = \|g_{k,1}\|$, meaning that the bound holds with $\rho = 1$. (This means that, in practice, one might choose $\rho \geq 1$ and iteratively remove columns of G_k for the computation of T_k until one finds $\|R_k^{-1}\| \leq \rho \|g_{k,1}\|^{-1}$, knowing that, in the extreme case, there will remain one column for which this condition is satisfied. However, for the reasons already given in Remark 2.2, we make Assumption 3.4, meaning that G_k always has m columns.)

We begin by stating two results that reveal important properties of the eigenvalues (corresponding to the elements of $\{T_k\}$) computed by the algorithm, which in turn reveal properties of the stepsizes. The first result is a direct consequence of the *Cauchy Interlacing Theorem*. Since this theorem is well-known—see, e.g., [4]—we state the lemma without proof.

Lemma 3.5. *For all $k \in \mathbb{N}$, the eigenvalues of T_k ($= Q_k^T A Q_k$ where $Q_k^T Q_k = I$) satisfy*

$$\theta_{k,j} \in [\lambda_{m+1-j}, \lambda_{n+1-j}] \quad \text{for all} \quad j \in [m].$$

The second result provides more detail about how the eigenvalues computed by the algorithm at the end of iteration $k \in \mathbb{N}$ relate to the weights in (9) corresponding to k for all $j \in [m]$.

Lemma 3.6. *For all $(k, j) \in \mathbb{N} \times [m]$, let $q_{k,j} \in \mathbb{R}^m$ denote the unit eigenvector corresponding to the eigenvalue $\theta_{k,j}$ of T_k, i.e., the vector satisfying $T_k q_{k,j} = \theta_{k,j} q_{k,j}$ and $\|q_{k,j}\| = 1$. Then, defining*

$$D_k := \begin{bmatrix} d_{k,1,1} & \cdots & d_{k,m,1} \\ \vdots & \ddots & \vdots \\ d_{k,1,n} & \cdots & d_{k,m,n} \end{bmatrix} \quad \text{and} \quad c_{k,j} := D_k R_k^{-1} q_{k,j}, \tag{12}$$

it follows that, with the diagonal matrix of eigenvalues (namely, Λ) defined in Assumption 2.1,

$$\theta_{k,j} = c_{k,j}^T \Lambda c_{k,j} \quad \text{and} \quad c_{k,j}^T c_{k,j} = 1. \tag{13}$$

Proof. For any $k \in \mathbb{N}$, it follows from (12) and Lemma 2.3 (in particular, (10)) that $G_k = Q D_k$ where Q is the orthogonal matrix defined in Assumption 2.1. Then, since $G_k = Q_k R_k$ (recall (4)), it follows that $Q_k = Q D_k R_k^{-1}$, according to which one finds

$$T_k = Q_k^T A Q_k = R_k^{-T} D_k^T Q^T A Q D_k R_k^{-1} = R_k^{-T} D_k^T \Lambda D_k R_k^{-1}.$$

Hence, for each $j \in [m]$, the first equation in (13) follows since

$$\theta_{k,j} = q_{k,j}^T T_k q_{k,j} = q_{k,j}^T R_k^{-T} D_k^T \Lambda D_k R_k^{-1} q_{k,j} = c_{k,j}^T \Lambda c_{k,j}.$$

In addition, since $G_k = Q D_k$ and the orthogonality of Q imply that $D_k^T D_k = G_k^T G_k$, and since $Q_k = G_k R_k^{-1}$ with Q_k having orthonormal columns (i.e., with Q_k satisfying $Q_k^T Q_k = I$), it follows that

$$c_{k,j}^T c_{k,j} = q_{k,j}^T R_k^{-T} D_k^T D_k R_k^{-1} q_{k,j} = q_{k,j}^T R_k^{-T} G_k^T G_k R_k^{-1} q_{k,j} = q_{k,j}^T Q_k^T Q_k q_{k,j} = q_{k,j}^T q_{k,j} = 1,$$

which yields the second equation in (13). □

The implications of Lemma 3.6 are seen later in our analysis. For now, combining Lemma 3.5, Lemma 2.3 (in particular, (10)), and the fact that (9) implies

$$\|g_{k,j}\|^2 = \sum_{i=1}^{n} d_{k,j,i}^2 \quad \text{for all} \quad (k, j) \in \mathbb{N} \times [m], \tag{14}$$

one is lead to the following result pertaining to recursive properties of the weights in (9).

Lemma 3.7. *For each $(k,j,i) \in \mathbb{N} \times [m] \times [n]$, it follows that*

$$|d_{k,j+1,i}| \leq \delta_{j,i}|d_{k,j,i}| \quad \textit{where} \quad \delta_{j,i} := \max\left\{\left|1 - \frac{\lambda_i}{\lambda_{m+1-j}}\right|, \left|1 - \frac{\lambda_i}{\lambda_{n+1-j}}\right|\right\}. \tag{15}$$

Hence, for each $(k,j,i) \in \mathbb{N} \times [m] \times [n]$, it follows that

$$|d_{k+1,j,i}| \leq \Delta_i |d_{k,j,i}| \quad \textit{where} \quad \Delta_i := \prod_{j=1}^{m} \delta_{j,i}. \tag{16}$$

Furthermore, for each $(k,j,p) \in \mathbb{N} \times [m] \times [n]$, it follows that

$$\sqrt{\sum_{i=1}^{p} d_{k,j+1,i}^2} \leq \hat{\delta}_{j,p} \sqrt{\sum_{i=1}^{p} d_{k,j,i}^2} \quad \textit{where} \quad \hat{\delta}_{j,p} := \max_{i \in [p]} \delta_{j,i}, \tag{17}$$

while, for each $(k,j) \in \mathbb{N} \times [m]$, it follows that

$$\|g_{k+1,j}\| \leq \Delta \|g_{k,j}\| \quad \textit{where} \quad \Delta := \max_{i \in [n]} \Delta_i. \tag{18}$$

Proof. Recall that, for any given $(k,j,i) \in \mathbb{N} \times [m] \times [n]$, Lemma 2.3 (in particular, (10)) states

$$d_{k,j+1,i} = (1 - \alpha_{k,j}\lambda_i) d_{k,j,i}.$$

The relationship (15) then follows due to Lemma 3.5, which, in particular, shows that

$$\alpha_{k,j} \in \left[\frac{1}{\lambda_{n+1-j}}, \frac{1}{\lambda_{m+1-j}}\right] \subseteq \left[\frac{1}{\lambda_n}, \frac{1}{\lambda_1}\right] \quad \text{for all} \quad (k,j) \in \mathbb{N} \times [m].$$

The consequence (16) then follows by combining (15) for all $j \in [m]$ and recalling that Step 11 yields $g_{k+1,1} \leftarrow g_{k,m+1}$ for all $k \in \mathbb{N}$. Now, from (15), one finds that

$$\sum_{i=1}^{p} d_{k,j+1,i}^2 \leq \sum_{i=1}^{p} \delta_{j,i}^2 d_{k,j,i}^2 \leq \hat{\delta}_{j,p}^2 \sum_{i=1}^{p} d_{k,j,i}^2 \quad \text{for all} \quad (k,j,p) \in \mathbb{N} \times [m] \times [n],$$

yielding the desired conclusion (17). Finally, combining (16) and (14), one obtains that

$$\|g_{k+1,j}\|^2 = \sum_{i=1}^{n} d_{k+1,j,i}^2 \leq \sum_{i=1}^{n} \Delta_i^2 d_{k,j,i}^2 \leq \Delta^2 \sum_{i=1}^{n} d_{k,j,i}^2 = \Delta^2 \|g_{k,j}\|^2 \quad \text{for all} \quad (k,j) \in \mathbb{N} \times [m],$$

yielding the desired conclusion (18). □

A consequence of the previous lemma is that if $\Delta_i \in [0,1)$ for all $i \in [n]$, then $\Delta \in [0,1)$, from which (18) implies that, for each $j \in [m]$, the gradient norm sequence $\{\|g_{k,j}\|\}_{k \in \mathbb{N}}$ vanishes Q-linearly. For example, such a situation occurs when $\lambda_n < 2\lambda_1$. However, as noted in [2], this is a highly special case that should not be assumed to hold widely in practice. A more interesting and widely relevant consequence of the lemma is that for any $i \in [n]$ such that $\Delta_i \in [0,1)$, the sequences $\{|d_{k,j,i}|\}_{k \in \mathbb{N}}$ for each $j \in [m]$ vanish Q-linearly. For example, this is *always* true for $i = 1$, where

$$\delta_{j,1} = \max\left\{1 - \frac{\lambda_1}{\lambda_{m+1-j}}, 1 - \frac{\lambda_1}{\lambda_{n+1-j}}\right\} \in [0,1) \quad \text{for all} \quad j \in [m],$$

from which it follows that

$$\Delta_1 = \prod_{j=1}^{m} \delta_{j,1} \in [0,1). \tag{19}$$

The following is a crucial consequence that one can draw from this observation.

Lemma 3.8. *If $\Delta_1 = 0$, then $d_{1+\hat{k},\hat{j},1} = 0$ for all $(\hat{k},\hat{j}) \in \mathbb{N} \times [m]$. Otherwise, if $\Delta_1 > 0$, then:*

(i) for any $(k,j) \in \mathbb{N} \times [m]$ such that $d_{k,j,1} = 0$, it follows that $d_{k+\hat{k},\hat{j},1} = 0$ for all $(\hat{k},\hat{j}) \in \mathbb{N} \times [m]$;

(ii) for any $(k,j) \in \mathbb{N} \times [m]$ such that $|d_{k,j,1}| > 0$ and any $\epsilon_1 \in (0,1)$, it follows that

$$\frac{|d_{k+\hat{k},\hat{j},1}|}{|d_{k,j,1}|} \leq \epsilon_1 \quad \text{for all} \quad \hat{k} \geq 1 + \left\lceil \frac{\log \epsilon_1}{\log \Delta_1} \right\rceil \quad \text{and} \quad \hat{j} \in [m].$$

Proof. If $\Delta_1 = 0$, then the desired conclusion follows from Lemma 3.7; in particular, it follows from the inequality (16) for $i = 1$. Similarly, for any $(k,j) \in \mathbb{N} \times [m]$ such that $d_{k,j,1} = 0$, the conclusion in part (i) follows from the same conclusion in Lemma 3.7, namely, (16) for $i = 1$. Hence, let us continue to prove part (ii) under the assumption that $\Delta_1 \in (0,1)$ (recall (19)).

Suppose that the given condition holds with $j = 1$, i.e., consider $k \in \mathbb{N}$ such that $|d_{k,1,1}| > 0$. Then, it follows by Lemma 3.7 (in particular, (16) for $j = 1$ and $i = 1$) that

$$\frac{|d_{k+\hat{k},1,1}|}{|d_{k,1,1}|} \leq \Delta_1^{\hat{k}} \quad \text{for any} \quad \hat{k} \in \mathbb{N}. \tag{20}$$

Since $\Delta_1 \in (0,1)$, taking the logarithm of the term on the right-hand side with $\hat{k} = \lceil \log \epsilon_1 / \log \Delta_1 \rceil$ yields

$$\left\lceil \frac{\log \epsilon_1}{\log \Delta_1} \right\rceil \log \Delta_1 \leq \left(\frac{\log \epsilon_1}{\log \Delta_1} \right) \log \Delta_1 = \log (\epsilon_1). \tag{21}$$

Since $\log(\cdot)$ is nondecreasing, the inequalities yielded by (21) combined with (20) along with (16) from Lemma 3.7 yield the desired result for $j = 1$. On the other hand, if the conditions of part (ii) hold for some other $j \in [m]$, then the desired conclusion follows from a similar reasoning, though an extra cycle may need to be completed before the desired conclusion holds for all points in the cycle, i.e., for all $\hat{j} \in [m]$; hence the addition of 1 to $\lceil \log \epsilon_1 / \log \Delta_1 \rceil$ in the general conclusion. □

One may conclude from Lemma 3.8 and (9) that, for any $(k,j) \in \mathbb{N} \times [m]$ and $\epsilon_1 \in (0,1)$, one has

$$\frac{|d_{k+\hat{k},\hat{j},1}|}{\|g_{k,j}\|} \leq \epsilon_1 \quad \text{for all} \quad \hat{k} \geq K_1 \quad \text{and} \quad \hat{j} \in [m]$$

for some $K_1 \in \mathbb{N}$ that depends on the desired contraction factor $\epsilon_1 \in (0,1)$ and the problem-dependent constant $\Delta_1 \in (0,1)$, but does *not* depend on the iteration number pair (k,j). Our goal now is to show that if a similar, but looser conclusion holds for a squared sum of the weights in (9) up through $p \in [n-1]$, then the squared weight corresponding to index $p+1$ eventually becomes sufficiently small in a number of iterations that is independent of the iteration number k. (For this lemma, we fix $j = \hat{j} = 1$ so as to consider only the first gradient in each cycle. This choice is somewhat arbitrary since our concluding theorem will confirm that a similar result would hold for any $j \in [m]$ and $\hat{j} = j$.) For the lemma, we define the following constants that dependent only on p, the spectrum of A (which, in particular, yields the bounds and definitions in (3.7)), and the scalar constant $\rho \geq 1$ from Assumption 3.4:

$$\hat{\delta}_p := \left(1 + \hat{\delta}_{1,p}^2 + \hat{\delta}_{1,p}^2 \hat{\delta}_{2,p}^2 + \cdots + \prod_{j=1}^{m-1} \hat{\delta}_{j,p}^2 \right) \in [1, \infty), \tag{22a}$$

$$\hat{\Delta}_{p+1} := \max \left\{ \frac{1}{3}, 1 - \frac{\lambda_{p+1}}{\lambda_n} \right\}^m \in (0,1), \tag{22b}$$

$$\text{and} \quad \hat{K}_p := \left\lceil \frac{\log \left(2\hat{\delta}_p \rho \epsilon_p \Delta_{p+1}^{-(K_p+1)} \right)}{\log \hat{\Delta}_{p+1}} \right\rceil. \tag{22c}$$

Lemma 3.9. *For any $(k,p) \in \mathbb{N} \times [n-1]$, if there exists $(\epsilon_p, K_p) \in (0, \frac{1}{2\hat{\delta}_p \rho}) \times \mathbb{N}$ independent of k with*

$$\sum_{i=1}^{p} d_{k+\hat{k},1,i}^2 \leq \epsilon_p^2 \|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k} \geq K_p, \tag{23}$$

then one of the following holds:

(i) $\Delta_{p+1} \in [0,1)$ and there exists $K_{p+1} \geq K_p$ dependent only on ϵ_p, ρ, and the spectrum of A with

$$d_{k+K_{p+1},1,p+1}^2 \leq 4\hat{\delta}_p^2 \rho^2 \epsilon_p^2 \|g_{k,1}\|^2; \tag{24}$$

(ii) $\Delta_{p+1} \in [1,\infty)$ and, with $K_{p+1} := K_p + \hat{K}_p + 1$, there exists $\hat{k}_0 \in \{K_p, \ldots, K_{p+1}\}$ with

$$d_{k+\hat{k}_0,1,p+1}^2 \leq 4\hat{\delta}_p^2 \rho^2 \epsilon_p^2 \|g_{k,1}\|^2. \tag{25}$$

Proof. By Lemma 3.7 (in particular, (16) with $j = 1$ and $i = p+1$) and (14), it follows that

$$d_{k+\hat{k},1,p+1}^2 \leq \left(\Delta_{p+1}^{\hat{k}} d_{k,1,p+1}\right)^2 = \Delta_{p+1}^{2\hat{k}} d_{k,1,p+1}^2 \leq \Delta_{p+1}^{2\hat{k}} \|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k} \in \mathbb{N}. \tag{26}$$

If $\Delta_{p+1} \in [0,1)$, then (26) immediately implies the existence of K_{p+1} dependent only on ϵ_p, ρ, and the spectrum of A such that (24) holds. Hence, let us continue under the assumption that $\Delta_{p+1} \geq 1$, where one should observe that $\rho \geq 1$, $\hat{\delta}_p \geq 1$, $\epsilon_p \in (0, \frac{1}{2\hat{\delta}_p \rho})$, $K_p \in \mathbb{N}$, and $\Delta_{p+1} \geq 1$ imply $2\hat{\delta}_p \rho \epsilon_p \Delta_{p+1}^{-K_p} \in (0,1)$, meaning that $\hat{K}_p \in \mathbb{N}$. To prove the desired result, it suffices to show that if

$$d_{k+\hat{k},1,p+1}^2 > 4\hat{\delta}_p^2 \rho^2 \epsilon_p^2 \|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k} \in \{K_p, \ldots, K_{p+1} - 1\}, \tag{27}$$

then (25) holds at the beginning of the next cycle (i.e., when $\hat{k}_0 = K_{p+1}$). From Lemma 3.6, Lemma 3.7 (in particular, (17)), (23), and (27), it follows that with $\{c_{k+\hat{k},j,i}\}_{i=1}^n$ representing the elements of the vector $c_{k+\hat{k},j}$ and the matrix $D_{k+\hat{k},p}$ representing the first p rows of $D_{k+\hat{k}}$, one finds

$$\begin{aligned}
\sum_{i=1}^{p} c_{k+\hat{k},j,i}^2 &\leq \|D_{k+\hat{k},p}\|_2^2 \|R_{k+\hat{k}}^{-1}\|^2 \|q_{k+\hat{k},j}\|^2 \\
&\leq \left(1 + \hat{\delta}_{1,p}^2 + \hat{\delta}_{1,p}^2 \hat{\delta}_{2,p}^2 + \cdots + \prod_{j=1}^{m-1} \hat{\delta}_{j,p}^2\right) \left(\sum_{i=1}^{p} d_{k+\hat{k},1,i}^2\right) \rho^2 \|g_{k+\hat{k},1}\|^{-2} \\
&\leq \hat{\delta}_p^2 (\epsilon_p^2 \|g_{k,1}\|^2) \rho^2 (4\hat{\delta}_p^2 \rho^2 \epsilon_p^2)^{-1} \|g_{k,1}\|^{-2} \leq \tfrac{1}{4} \quad \text{for all} \quad \hat{k} \in \{K_p, \ldots, K_{p+1} - 1\} \quad \text{and} \quad j \in [m].
\end{aligned}$$

Along with Lemma 3.6, this implies that

$$\theta_{k+\hat{k},j} = \sum_{i=1}^{n} \lambda_i c_{k+\hat{k},j,1}^2 \geq \tfrac{3}{4} \lambda_{p+1} \quad \text{for all} \quad \hat{k} \in \{K_p, \ldots, K_{p+1} - 1\} \quad \text{and} \quad j \in [m]. \tag{28}$$

Together with Lemma 2.3 (see (10)) and $\alpha_{k+\hat{k}+1,j} = \theta_{k+\hat{k},j}^{-1}$ for all $j \in [m]$, the bound (28) implies

$$\begin{aligned}
d_{k+\hat{k}+2,1,p+1}^2 &= \left(\prod_{j=1}^{m} \left(1 - \alpha_{k+\hat{k}+1,j} \lambda_{p+1}\right)^2\right) d_{k+\hat{k}+1,1,p+1}^2 \\
&\leq \hat{\Delta}_{p+1}^2 d_{k+\hat{k}+1,1,p+1}^2 \quad \text{for all} \quad \hat{k} \in \{K_p, \ldots, K_{p+1} - 1\}.
\end{aligned} \tag{29}$$

Applying this bound recursively, it follows with $K_{p+1} = K_p + \hat{K}_p + 1$ and (26) for $\hat{k} = K_{p+1}$ that

$$d_{k+K_{p+1},1,p+1}^2 \leq \hat{\Delta}_{p+1}^{2\hat{K}_p} d_{k+K_p+1,1,p+1}^2 \leq \hat{\Delta}_{p+1}^{2\hat{K}_p} \Delta_{p+1}^{2(K_p+1)} \|g_{k,1}\|^2 \leq 4\hat{\delta}_p^2 r^2 \epsilon_p^2 \|g_{k,1}\|^2,$$

where the last inequality follows by the definition of $\hat{K}_p$ in (22c). □

We have shown that small squared weights in (9) associated with indices up through $p \in [n-1]$ imply that the squared weight associated with index $p+1$ eventually becomes small. The next lemma shows that these latter squared weights also remain sufficiently small indefinitely.

Lemma 3.10. *For any $(k,p) \in \mathbb{N} \times [n-1]$, if there exists $(\epsilon_p, K_p) \in (0, \frac{1}{2\hat{\delta}_p \rho}) \times \mathbb{N}$ independent of k such that (23) holds, then, with $\epsilon_{p+1}^2 := (1 + 4\max\{1, \Delta_{p+1}^4\}\hat{\delta}_p^2\rho^2)\epsilon_p^2$ and $K_{p+1} \in \mathbb{N}$ from Lemma 3.9,*

$$\sum_{i=1}^{p+1} d_{k+\hat{k},1,i}^2 \leq \epsilon_{p+1}^2 \|g_{k,1}\|^2 \quad \textit{for all} \quad \hat{k} \geq K_{p+1}. \tag{30}$$

Proof. For the same reasons as in the proof of Lemma 3.9, the result follows if $\Delta_{p+1} \in [0,1)$. Hence, we may continue under the assumption that $\Delta_{p+1} \geq 1$ and define $\hat{\Delta}_{p+1} \in (0,1)$ and $\hat{K}_p \in \mathbb{N}$ as in (22). By Lemma 3.9, there exists $\hat{k}_0 \in \{K_p, \ldots, K_{p+1}\}$ such that

$$d_{k+\hat{k},1,p+1}^2 \leq 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \quad \text{when} \quad \hat{k} = \hat{k}_0. \tag{31}$$

If the inequality in (31) holds for all $\hat{k} \geq \hat{k}_0$, then (30) holds with $\epsilon_{p+1}^2 = (1+4\hat{\delta}_p^2\rho^2)\epsilon_p^2$. Otherwise, let $\hat{k}_1 \in \mathbb{N}$ denote the smallest natural number such that

$$d_{k+\hat{k},1,p+1}^2 \leq 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k}_0 \leq \hat{k} \leq \hat{k}_1, \tag{32}$$

but

$$d_{k+\hat{k}_1+1,1,p+1}^2 > 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2. \tag{33}$$

As in the arguments that lead to (29) in the proof of Lemma 3.9, combining (23) and (33) implies

$$d_{k+\hat{k}_1+3,1,p+1}^2 \leq \hat{\Delta}_{p+1}^2 d_{k+\hat{k}_1+2,1,p+1}^2.$$

Generally, this same argument can be used to show that

$$\hat{k} \geq K_p \quad \text{and} \quad d_{k+\hat{k}+1,1,p+1}^2 > 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \quad \text{imply} \quad d_{k+\hat{k}+3,1,p+1}^2 \leq \hat{\Delta}_{p+1}^2 d_{k+\hat{k}+2,1,p+1}^2.$$

Since $\hat{\Delta}_{p+1} \in (0,1)$, this fact and (33) imply the existence of $\hat{k}_2 \in \mathbb{N}$ such that

$$d_{k+\hat{k}+1,1,p+1}^2 > 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k}_1 \leq \hat{k} \leq \hat{k}_2 - 2, \tag{34}$$

but

$$d_{k+\hat{k}_2,1,p+1}^2 \leq 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2,$$

while, from above,

$$d_{k+\hat{k}+3,1,p+1}^2 \leq \hat{\Delta}_{p+1}^2 d_{k+\hat{k}+2,1,p+1}^2 \quad \text{for all} \quad \hat{k}_1 \leq \hat{k} \leq \hat{k}_2 - 2. \tag{35}$$

Moreover, by Lemma 3.7 (in particular, (16)) and (32), it follows that

$$d_{k+\hat{k}_1+1,1,p+1}^2 \leq \Delta_{p+1}^2 d_{k+\hat{k}_1,1,p+1}^2 \leq 4\Delta_{p+1}^2\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \tag{36a}$$

$$\text{and} \quad d_{k+\hat{k}_1+2,1,p+1}^2 \leq 4\Delta_{p+1}^4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2. \tag{36b}$$

Combining (35) and (36b), it follows that

$$d_{k+\hat{k}+3,1,p+1}^2 \leq 4\hat{\Delta}_{p+1}^2\Delta_{p+1}^4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k}_1 \leq \hat{k} \leq \hat{k}_2 - 2.$$

Overall, since (22b) ensures $\hat{\Delta}_{p+1} \in (0,1)$, we have shown that

$$d_{k+\hat{k},1,p+1}^2 \leq 4\Delta_{p+1}^4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k} \in \{\hat{k}_0, \ldots, \hat{k}_2\}. \tag{37}$$

Repeating this argument for later iterations, we arrive at the desired conclusion. □

The following lemma is a generalization of Lemma 3.2 for any $m \in \mathbb{N}$. Our proof is similar to that of Lemma 2.4 in [2]. We provide it in full for completeness.

Lemma 3.11. *There exists $K \in \mathbb{N}$ dependent only on the spectrum of A such that*

$$\|g_{k+K,1}\| \leq \tfrac{1}{2}\|g_{k,1}\| \quad \textit{for all} \quad k \in \mathbb{N}.$$

Proof. By Lemma 3.10, if for some $(\epsilon_p, K_p) \in (0, \frac{1}{2\hat{\delta}_p \rho}) \times \mathbb{N}$ independent of k one finds

$$\sum_{i=1}^{p} d_{k+\hat{k},1,i}^2 \leq \epsilon_p^2 \|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k} \geq K_p, \tag{38}$$

then for $\epsilon_{p+1}^2 := (1 + 4\max\{1, \Delta_{p+1}^4\}\hat{\delta}_p^2 \rho^2)\epsilon_p^2$ and some $K_{p+1} \geq K_p$ independent of k one finds

$$\sum_{i=1}^{p+1} d_{k+\hat{k},1,i}^2 \leq \epsilon_{p+1}^2 \|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k} \geq K_{p+1}. \tag{39}$$

Since Lemma 3.8 implies that for any $\epsilon_1 \in (0,1)$ one can find K_1 independent of k such that (38) holds with $p = 1$, it follows that, independent of k, there exists a sufficiently small $\epsilon_1 \in (0,1)$ such that

$$\epsilon_1^2 \leq \cdots \leq \epsilon_n^2 \leq \tfrac{1}{4}.$$

Hence, for any $k \in \mathbb{N}$, it follows that there exists $K = K_n$ such that

$$\|g_{k+\hat{k},1}\|^2 = \sum_{i=1}^{n} d_{k+\hat{k},1,i}^2 \leq \tfrac{1}{4}\|g_{k,1}\|^2 \quad \text{for all} \quad \hat{k} \geq K,$$

as desired. □

We are now prepared to state our final result, the proof of which follows in the same manner as Theorem 3.3 follows from Lemma 3.2 in [2]. We prove it in full for completeness.

Theorem 3.12. *The sequence $\{\|g_{k,1}\|\}$ vanishes R-linearly.*

Proof. If $\Delta \in [0,1)$, then it has already been argued (see the discussion following Lemma 3.7) that $\{\|g_{k,1}\|\}$ vanishes Q-linearly. Hence, let us continue assuming that $\Delta \geq 1$. By Lemma 3.11, there exists $K \in \mathbb{N}$ dependent only on the spectrum of A such that

$$\|g_{1+Kl,1}\| \leq \tfrac{1}{2}\|g_{1+K(l-1),1}\| \quad \text{for all} \quad l \in \mathbb{N}.$$

Applying this result recursively, it follows that

$$\|g_{1+Kl,1}\| \leq (\tfrac{1}{2})^l \|g_{1,1}\| \quad \text{for all} \quad l \in \mathbb{N}. \tag{40}$$

Now, for any $k \geq 1$, let us write $k = Kl + \hat{k}$ for some $l \in \{0\} \cup \mathbb{N}$ and $\hat{k} \in \{0\} \cup [K-1]$. It follows that

$$l = k/K - \hat{k}/K \geq k/K - 1.$$

By this fact, (18), and (40), it follows that for any $k = Kl + \hat{k} \in \mathbb{N}$ one has

$$\|g_{k,1}\| \leq \Delta^{\hat{k}-1}\|g_{1+Kl,1}\| \leq \Delta^{K-1}(\tfrac{1}{2})^{k/K-1}\|g_{1,1}\| \leq c_1 c_2^k \|g_{1,1}\|,$$

where

$$c_1 := 2\Delta^{K-1} \quad \text{and} \quad c_2 := 2^{-1/K} \in (0,1),$$

which implies the desired conclusion. □

4 Numerical Demonstrations

The analysis in the previous section provides additional insights into the behavior of Algorithm LMSD beyond its R-linear rate of convergence. In this section, we provide the results of numerical experiments to demonstrate the behavior of the algorithm in a few types of cases. The algorithm was implemented and the experiments were performed in Matlab. It is not our goal to show the performance of Algorithm LMSD for various values of m, say to argue whether the performance improves or not as m is increased. This is an important question for which some interesting discussion is provided by [3]. However, to determine what is a good choice of m for various types of cases would require a larger set of experiments that are outside of the scope of this paper. For our purposes, our only goal is to provide some simple illustrations of the behavior as shown by our theoretical analysis.

Our analysis reveals that the convergence behavior of the algorithm depends on the spectrum of the matrix A. Therefore, we have constructed five test examples, all with $n = 100$, but with different eigenvalue distributions. For the first problem, the eigenvalues of A are evenly distributed in $[1, 1.9]$. Since this ensures that $\lambda_n < 2\lambda_1$, our analysis reveals that the algorithm converges Q-linearly for this problem; recall the discussion after Lemma 3.7. All other problems were constructed so that $\lambda_1 = 1$ and $\lambda_n = 100$, for which one clearly finds $\lambda_n > 2\lambda_1$. For the second problem, all eigenvalues are evenly distributed in $[\lambda_1, \lambda_n]$; for the third problem, the eigenvalues are clustered in five distinct blocks; for the fourth problem, all eigenvalues except one are clustered around λ_1; and for the fifth problem, all eigenvalues except one are clustered around λ_n. Table 1 shows the spectrum of A for each problem.

The table also shows the numbers of outer and (total) inner iterations required by Algorithm LMSD (indicated by column headers "k" and "j", respectively) when it was run with $\epsilon = 10^{-8}$ and either $m = 1$ or $m = 5$. In all cases, the initial m stepsizes were generated randomly from a uniform distribution over the interval $[\lambda_{100}^{-1}, \lambda_1^{-1}]$. One finds that the algorithm terminates in relatively few outer and inner iterations relative to n, especially when many of the eigenvalues are clustered. This dependence on clustering of the eigenvalues should not be surprising since, recalling Lemma 3.5, clustered eigenvalues makes it likely that an eigenvalue of T_k will be near an eigenvalue of A, which in turn implies by Lemma 2.3 that the weights in the representation (9) will vanish quickly. On the other hand, for the problems for which the eigenvalues are more evenly spread in $[1, 100]$, the algorithm requires relatively more outer iterations, though still not an excessively large number relative to n. For these problems, the performance was better for $m = 5$ versus $m = 1$, both in terms of outer and (total) inner iterations.

Table 1: Spectra of A for five test problems along with outer and (total) inner iteration counts required by Algorithm LMSD. For each spectrum, a set of eigenvalues in an interval indicates that the eigenvalues are evenly distributed within that interval.

Problem	Spectrum			$m = 1$ k	$m = 1$ j	$m = 5$ k	$m = 5$ j
1	$\{\lambda_1, \ldots, \lambda_{100}\}$	$\subset$	$[1, 1.9]$	13	13	3	14
2	$\{\lambda_1, \ldots, \lambda_{100}\}$	$\subset$	$[1, 100]$	124	124	23	114
3	$\{\lambda_1, \ldots, \lambda_{20}\}$	$\subset$	$[1, 2]$	112	112	16	79
	$\{\lambda_{21}, \ldots, \lambda_{40}\}$	$\subset$	$[25, 26]$				
	$\{\lambda_{41}, \ldots, \lambda_{60}\}$	$\subset$	$[50, 51]$				
	$\{\lambda_{61}, \ldots, \lambda_{80}\}$	$\subset$	$[75, 76]$				
	$\{\lambda_{81}, \ldots, \lambda_{100}\}$	$\subset$	$[99, 100]$				
4	$\{\lambda_1, \ldots, \lambda_{99}\}$	$\subset$	$[1, 2]$	26	26	4	20
	λ_{100}	$=$	100				
5	λ_1	$=$	1	16	16	5	25
	$\{\lambda_2, \ldots, \lambda_{100}\}$	$\subset$	$[99, 100]$				

As seen in our analysis (inspired by [5], [2], and [3]), a more refined look into the behavior of the algorithm is obtained by observing the step-by-step magnitudes of the weights in (9) for the generated gradients. Hence,

for each of the test problems, we plot in Figures 1, 3, 5, 7, and 9 these magnitudes (on a log scale) for a few representative values of $i \in [n]$. Each figure consists of four sets of plots: the first and third show the magnitudes corresponding to $\{g_{k,1}\}$ (i.e., for the first point in each cycle) when $m = 1$ and $m = 5$, respectively, while the second and fourth show the magnitudes at all outer and inner iterations, again when $m = 1$ and $m = 5$, respectively. In a few of the images, the plot ends before the right-hand edge of the image. This is due to the log of the absolute value of the weight being evaluated as $-\infty$ in Matlab.

The tables show that the magnitudes of the weights corresponding to $i = 1$ always decrease monotonically, as proved in Lemma 3.8. The magnitudes corresponding to $i = 2$ also often decrease monotonically, but, as seen in the results for Problem 5, this is not always the case. In any case, the magnitudes corresponding to $i = 50$ and $i = 100$ often do not decrease monotonically, though, as proved in our analysis, one observes that the magnitudes demonstrate a downward trend over a finite number of cycles.

Even further insight into the plots of these magnitudes can be gained by observing the values of the constants $\{\Delta_i\}_{i\in[n]}$ for each problem and history length. Recalling (16), these constants bound the increase that a particular weight in (9) might experience from one point in a cycle to the same point in the subsequent cycle. For illustration, we plot in Figures 2, 4, 6, 8, and 10 these constants. Values less than 1 are indicated by a purple bar while values greater than or equal to 1 are indicated by a blue bar. Note that, in Figure 8, all values are small for both history lengths except Δ_{100}. In Figure 10, Δ_1 is less than one in both figures, but the remaining constants are large for $m = 1$ while being small for $m = 5$.

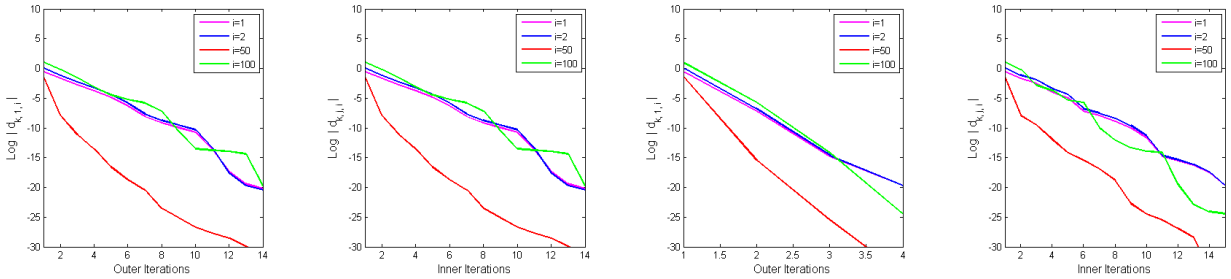

Figure 1: Weights in (9) for problem 1 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

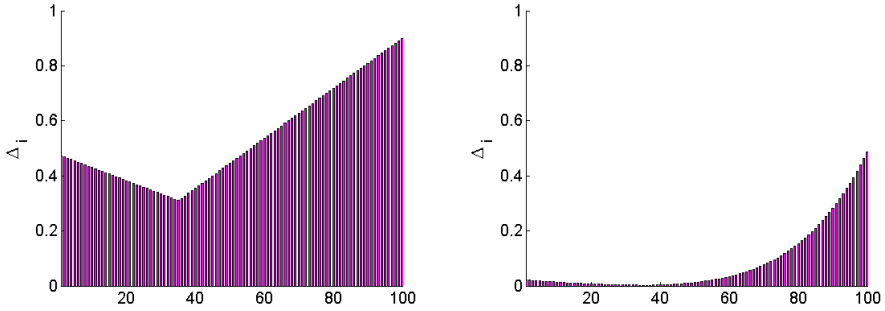

Figure 2: Constants in (16) for problem 1 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

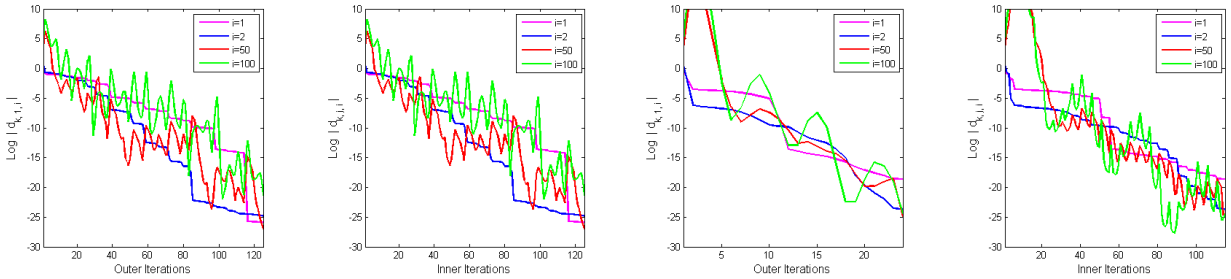

Figure 3: Weights in (9) for problem 2 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

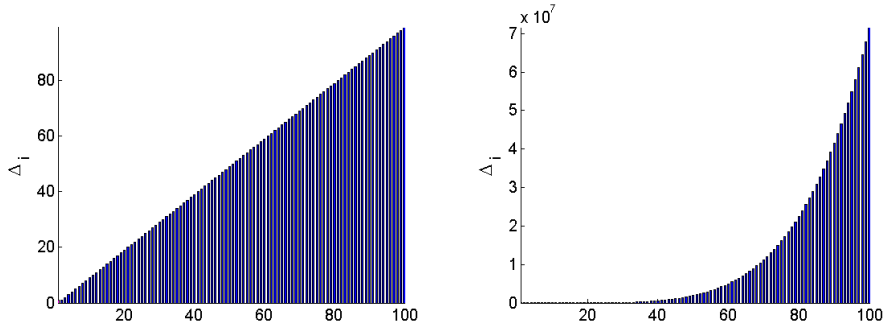

Figure 4: Constants in (16) for problem 2 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

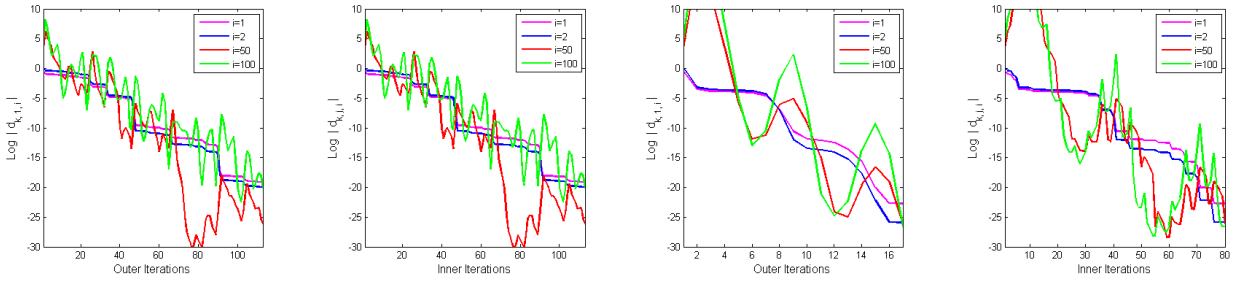

Figure 5: Weights in (9) for problem 3 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

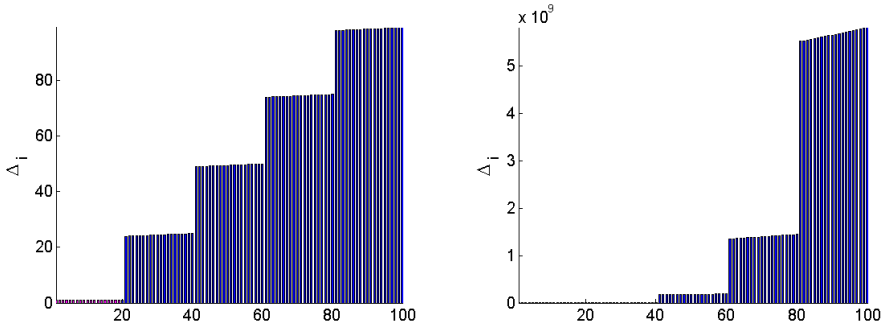

Figure 6: Constants in (16) for problem 3 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

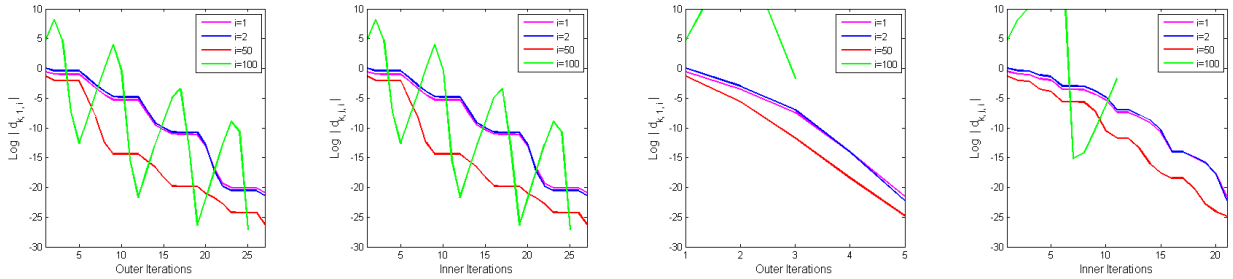

Figure 7: Weights in (9) for problem 4 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

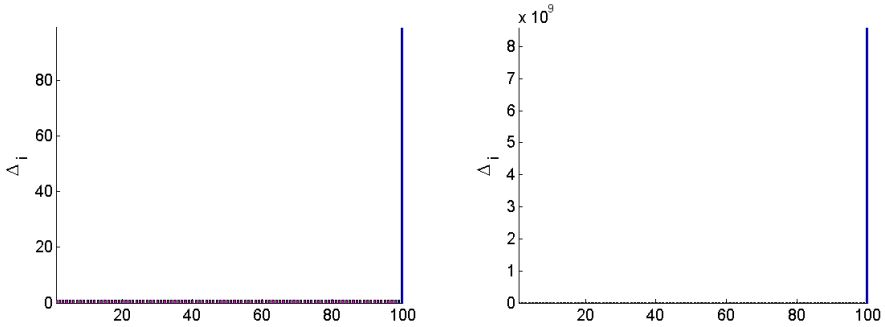

Figure 8: Constants in (16) for problem 4 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

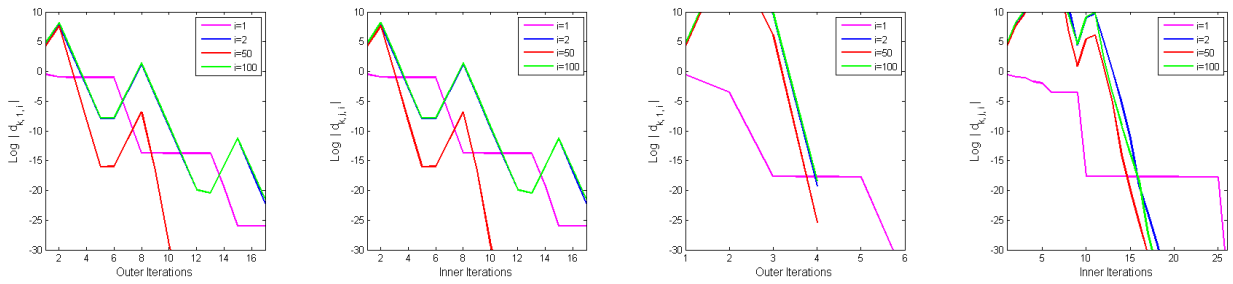

Figure 9: Weights in (9) for problem 5 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

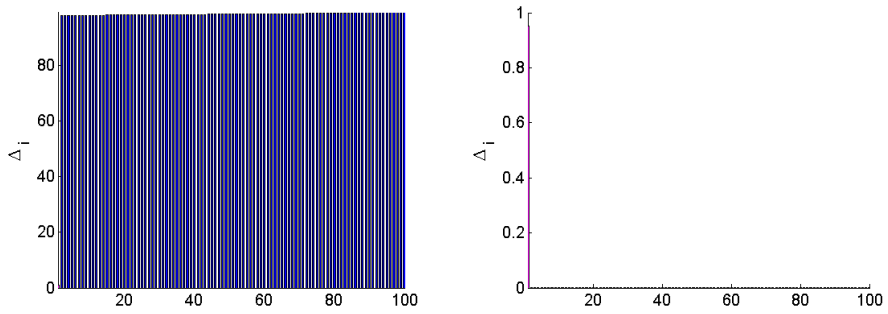

Figure 10: Constants in (16) for problem 5 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

5 Conclusion

We have shown that the limited memory steepest descent (LMSD) method proposed by [3] possesses an R-linear rate of convergence for any history length $m \in \mathbb{N}$ when it is employed to minimize a strongly convex quadratic function. Our analysis effectively extends that in [2], which covers only the $m = 1$ case. We have also provided the results of numerical experiments to demonstrate the practical performance of the algorithm, the results of which are informed by our theoretical analysis.

References

[1] J. Barzilai and J. M. Borwein. Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[2] Y.-H. Dai and L.-Z. Liao. R-linear Convergence of the Barzilai and Borwein Gradient Method. *IMA Journal of Numerical Analysis*, 22:1–10, 2002.

[3] R. Fletcher. A Limited Memory Steepest Descent Method. *Mathematical Programming*, 135(1-2):413–436, 2012.

[4] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.

[5] M. Raydan. On the Barzilai and Borwein Choice of Steplength for the Gradient Method. *IMA Journal of Numerical Analysis*, 13(3):321–326, 1993.