IMA Journal of Numerical Analysis (2018) **38**, 720–742
doi: 10.1093/imanum/drx016
Advance Access publication on April 24, 2017

R-linear convergence of limited memory steepest descent

Frank E. Curtis[*] and Wei Guo
Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA
[*]Corresponding author: frank.e.curtis@gmail.com weg411@lehigh.edu

[Received on 12 October 2016; revised on 5 February 2017]

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

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 article extends this analysis for LMSD, also for strongly convex quadratics. In particular, it is shown that, under reasonable assumptions, the method is *R*-linearly convergent for any choice of the history length parameter. The results of numerical experiments are also 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.

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 Barzilai & Borwein (1988), where a 'two-point step size' strategy is proposed and analysed. 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 step size 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 for any smooth objective function. In the original work (i.e., Barzilai & Borwein, 1988), it is shown that the method converges R-superlinearly when $n = 2$. In Raydan (1993), it is shown that the method converges from any starting point for any natural number n, and in Dai & Liao (2002), it is shown that the method converges R-linearly for any such n.

In each iteration of the BB method, the step size 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 Fletcher (2012), this idea can be extended to a *limited memory steepest descent* (LMSD) method in which a *sequence* of m step sizes 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 step size choices. Indeed, if the eigenvalues are computed exactly, then the algorithm terminates in a finite number of iterations; e.g., see Lai (1981), Fletcher (2012) and Section 2.

In Fletcher (2012), 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 analysed. The main purpose of this article is to show that, under reasonable assumptions, this LMSD method converges *R*-linearly when employed to minimize such a function. Our analysis builds upon the analyses in Fletcher (2012) and Dai & Liao (2002).

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

Notation

The set of real numbers (i.e., scalars) is denoted as $\mathbb{R}$, the set of non-negative 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 article. We then state Fletcher's LMSD algorithm and prove a finite termination property for it, as is done in Lai (1981) and Fletcher (2012).

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,

$$\min_{x\in\mathbb{R}^n} f(x), \quad \text{where } f(x) = \tfrac{1}{2}x^TAx - b^Tx. \tag{2.1}$$

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

Assumption 2.1 The matrix A in problem (2.1) has $r \le n$ distinct eigenvalues denoted by

$$\lambda_{(r)} > \cdots > \lambda_{(1)} > 0. \tag{2.2}$$

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

$$\begin{aligned} Q &= \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix} && \text{is orthogonal} \\ \text{and } \Lambda &= \operatorname{diag}(\lambda_1,\ldots,\lambda_n) && \text{with } \lambda_n \ge \cdots \ge \lambda_1 > 0 \\ & && \text{and } \lambda_i \in \{\lambda_{(1)},\ldots,\lambda_{(r)}\} \text{ for all } i \in [n]. \end{aligned} \tag{2.3}$$

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 analyse 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 *LMSD method*

Fletcher's LMSD 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 step sizes $\{\alpha_{k,j}\}_{j\in[m]}$ 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 step sizes to be employed in the next cycle are computed as the reciprocals of Ritz values of A, i.e., estimates of eigenvalues of A that are contained in the spectrum of A in a certain desirable sense (e.g., see Lemma 3.6 in Section 3.2). Fletcher (2012) 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 $T_k := Q_k^TAQ_k$, where $Q_k \in \mathbb{R}^{n\times m}$ satisfying $Q_k^TQ_k = I$ 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

$$Q_kR_k = G_k := \begin{bmatrix} g_{k,1} & \cdots & g_{k,m} \end{bmatrix}. \tag{2.4}$$

(For now, let us assume that G_k has linearly independent columns, in which case the matrix R_k in (2.4) 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 storage of the n-vectors composing the columns of Q_k. Both can be avoided in the following manner. First, it can be shown from the iterate update in Step 7 of Algorithm LMSD (e.g., see the proof of Lemma 2.3 in Section 2.3) that $g_{k,j+1} = g_{k,j} - \alpha_{k,j} A g_{k,j}$, for all $(k,j) \in \mathbb{N} \times [m]$. This means that, with the gradient at the initial point of cycle $k+1$, namely $g_{k+1,1} \equiv g_{k,m+1}$, and the matrix of kth-cycle reciprocal step sizes, namely,

$$J_k \leftarrow \begin{bmatrix} \alpha_{k,1}^{-1} & & \\ -\alpha_{k,1}^{-1} & \ddots & \\ & \ddots & \alpha_{k,m}^{-1} \\ & & -\alpha_{k,m}^{-1} \end{bmatrix}, \tag{2.5}$$

one has $AG_k = \begin{bmatrix} G_k & g_{k,m+1} \end{bmatrix} J_k$, which in turn means that

$$G_k^T A G_k = G_k^T \begin{bmatrix} G_k & g_{k,m+1} \end{bmatrix} J_k. \tag{2.6}$$

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

$$G_k^T \begin{bmatrix} G_k & g_{k,m+1} \end{bmatrix} = R_k^T \begin{bmatrix} R_k & r_k \end{bmatrix}, \tag{2.7}$$

one can see [by plugging (2.7) into (2.6) and using $G_k = Q_k R_k$] that T_k can be computed by

$$T_k \leftarrow \begin{bmatrix} R_k & r_k \end{bmatrix} J_k R_k^{-1}. \tag{2.8}$$

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

$$T_k \leftarrow \begin{bmatrix} R_k & Q_k^T g_{k,m+1} \end{bmatrix} J_k R_k^{-1}. \tag{2.9}$$

However, this is less efficient than using (2.8) 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 (2.8) (see Fletcher, 2012).

The choice to order the eigenvalues of T_k in decreasing order is motivated by Fletcher (2012). In short, this ensures that the step sizes 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 step sizes $\{\alpha_{1,j}\}_{j \in [m]} \subset \mathbb{R}_{++}$. This choice has no effect on the theoretical results proved in this article, though our analysis does confirm the fact that the practical performance of the method can be improved if one has the knowledge to choose one or more step sizes exactly equal to reciprocals of eigenvalues of A; see Section 2.3. Otherwise, one can either provide a full set of m step sizes 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 Fletcher (2012), for further discussion on this matter.

REMARK 2.2 In (2.4), if G_k for some $k \in \mathbb{N}$ does not have linearly independent columns, then R_k is singular and the formulas (2.8) and (2.9) are invalid, meaning that the employed approach is not able to provide m eigenvalue estimates for cycle k. As suggested in Fletcher (2012), an implementation of the

Algorithm LMSD Limited Memory Steepest Descent Method

```
choose an initial point $x_{1,1} \in \mathbb{R}^n$, history length $m \in [n]$, 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 (2.4) and $J_k$ by (2.5)
    compute $R_k$ and $r_k$ to satisfy (2.7) and set $T_k$ by (2.8)
    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
```

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 Section 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 Section 3.

2.3 *Finite termination property of LMSD*

If, for some $k \in \mathbb{N}$ and $j \in [m]$, the step sizes 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, e.g., in Fletcher (2012, Section 2).

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 (2.3), i.e.,

$$g_{k,j} = \sum_{i=1}^{n} d_{k,j,i} q_i. \tag{2.10}$$

Moreover, these weights satisfy the recursive property

$$d_{k,j+1,i} = (1 - \alpha_{k,j}\lambda_i)d_{k,j,i} \text{ for all } (k,j,i) \in \mathbb{N} \times [m] \times [n]. \tag{2.11}$$

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

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

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 step sizes 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 step size 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

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

Along with the facts that Lemma 2.3 also implies

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

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 $m \in [n]$. For context, we begin by citing two known convergence results that apply to Algorithm LMSD, then turn our attention to our new convergence rate results.

3.1 *Known convergence properties of LMSD*

In the appendix of Fletcher (2012), 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 (2.10), the following limits hold:

$$\lim_{k\to\infty} g_{k,j} = 0 \;\text{ for each }\; j \in [m] \;\text{ and} \tag{3.1a}$$

$$\lim_{k\to\infty} d_{k,j,i} = 0 \;\text{ for each }\; (j,i) \in [m] \times [n]. \tag{3.1b}$$

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 does the following results from Dai & Liao (2002) (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

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

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

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

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

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

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 $m \in [n]$*

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, under reasonable assumptions, Algorithm LMSD possesses an R-linear rate of convergence for any $m \in [n]$. 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. One of the main challenges in this pursuit is the fact, hinted at by Lemma 3.2 for the case of $m = 1$, that the gradients computed in Algorithm LMSD might not decrease monotonically in norm. This is one reason why the analysis in Dai & Liao (2002) is so remarkable, and, not surprisingly, it is an issue that must be overcome in our analysis as well. But our analysis also overcomes new challenges. In particular, the analysis in Dai & Liao (2002) is able to be more straightforward due to the fact that, in a BB method, a step size computation is performed after *every* iterate update. In particular, this means that, in iteration $k \in \mathbb{N}$, the current gradient $g_{k,1}$ plays a role in the computation of $\alpha_{k,1}$. In LMSD, on the other hand, a set of step sizes are computed and employed in sequence, meaning that multiple iterate updates are performed until the next set of step sizes are computed. This means that, in each cycle, iterate updates are performed using step sizes computed using *old* gradient information. Another challenge that our analysis overcomes is the fact that the computed step sizes cannot all be characterized in the same manner; rather, as revealed later in Lemma 3.7, each set of step sizes is *spread* through distinct intervals in the spectrum of A. Our analysis overcomes all these challenges by keeping careful track of the affects of applying each sequence of step sizes vis-à-vis the weights in (2.10) for all $(k,j) \in \mathbb{N} \times [m]$. In particular, we show that even though the gradients might not decrease monotonically in norm and certain weights in (2.10) might increase within each cycle, and from one cycle to the next, the weights ultimately vanish in a manner that corresponds to R-linear vanishing of the gradients for any $m \in [n]$.

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 because, 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.)

REMARK 3.5 Our analysis hinges on properties of the step sizes computed in Steps 12–15 of Algorithm LMSD as they relate to the spectrum of the matrix A. These properties do not necessarily hold for the initial set of step sizes $\{\alpha_{1,j}\}_{j \in [m]}$, which are merely restricted to be in $\mathbb{R}_{++}$. However, for ease of exposition in our analysis, rather than distinguish between the step sizes in the initial cycle (i.e., $k = 1$) vs. all subsequent cycles (i.e., $k \geq 2$), we proceed under the assumption that all properties that hold for $k \geq 2$ also hold for $k = 1$. (One could instead imagine that an 'initialization cycle' is performed corresponding to $k = 0$, in which case all of our subsequent results are indeed true for all $k \in \mathbb{N}$.) We proceed in this manner, without stating it as a formal assumption, since our main conclusion (see Theorem 3.13) remains true whether or not one counts the computational effort in the initial cycle.

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 step sizes. The first result is a direct consequence of the *Cauchy interlacing theorem*. Since this theorem is well-known—see, e.g., Parlett (1998)—we state the lemma without proof.

LEMMA 3.6 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

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

The second result provides more details about how the eigenvalues computed by the algorithm at the end of iteration $k \in \mathbb{N}$ relate to the weights in (2.10) corresponding to k for all $j \in [m]$.

LEMMA 3.7 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

$$D_k := \begin{bmatrix} d_{k,1,1} & \cdots & d_{k,m,1} \\ \vdots & \ddots & \vdots \\ d_{k,1,n} & \cdots & d_{k,m,n} \end{bmatrix} \text{ and } c_{k,j} := D_k R_k^{-1} q_{k,j}, \tag{3.2}$$

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

$$\theta_{k,j} = c_{k,j}^T \Lambda c_{k,j} \text{ and } c_{k,j}^T c_{k,j} = 1. \tag{3.3}$$

Proof. For any $k \in \mathbb{N}$, it follows from (3.2) and Lemma 2.3 [in particular, (2.11)] that $G_k = QD_k$, where Q is the orthogonal matrix defined in Assumption 2.1. Then, since $G_k = Q_k R_k$ [recall (2.4)], it follows that $Q_k = QD_k R_k^{-1}$, 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 (3.3) 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 $G_k = QD_k$ 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 (3.3). □

The implications for Lemma 3.7 are seen later in our analysis. For now, combining Lemma 3.6, Lemma 2.3 [in particular, (2.11)] and the fact that (2.10) implies

$$\|g_{k,j}\|^2 = \sum_{i=1}^{n} d_{k,j,i}^2 \text{ for all } (k,j) \in \mathbb{N} \times [m], \tag{3.4}$$

one is led to the following result pertaining to recursive properties of the weights in (2.10).

LEMMA 3.8 For each $(k,j,i) \in \mathbb{N} \times [m] \times [n]$, it follows that

$$|d_{k,j+1,i}| \le \delta_{j,i}|d_{k,j,i}|, \text{ where } \delta_{j,i} := \max\left\{\left|1 - \frac{\lambda_i}{\lambda_{m+1-j}}\right|, \left|1 - \frac{\lambda_i}{\lambda_{n+1-j}}\right|\right\}. \tag{3.5}$$

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

$$|d_{k+1,j,i}| \le \Delta_i|d_{k,j,i}|, \text{ where } \Delta_i := \prod_{j=1}^{m} \delta_{j,i}. \tag{3.6}$$

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

$$\sqrt{\sum_{i=1}^{p} d_{k,j+1,i}^2} \le \hat{\delta}_{j,p}\sqrt{\sum_{i=1}^{p} d_{k,j,i}^2}, \text{ where } \hat{\delta}_{j,p} := \max_{i\in[p]} \delta_{j,i}, \tag{3.7}$$

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

$$\|g_{k+1,j}\| \le \Delta\|g_{k,j}\|, \text{ where } \Delta := \max_{i\in[n]} \Delta_i. \tag{3.8}$$

Proof. Recall that, for any given $(k,j,i) \in \mathbb{N} \times [m] \times [n]$, Lemma 2.3 [in particular, (2.11)] states

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

The relationship (3.5) then follows due to Lemma 3.6, which, in particular, shows that

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

The consequence (3.6) then follows by combining (3.5) 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 (3.5), one finds that

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

yielding the desired conclusion (3.7). Finally, combining (3.6) and (3.4), one obtains that

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

yielding the desired conclusion (3.8). □

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 (3.8) 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 Dai & Liao (2002), this is a highly uncommon 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

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

from which it follows that

$$\Delta_1 = \prod_{j=1}^{m} \delta_{j,1} \in [0,1). \tag{3.9}$$

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

LEMMA 3.9 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

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

Proof. If $\Delta_1 = 0$, then the desired conclusion follows from Lemma 3.8; in particular, it follows from the inequality (3.6) 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 3.8, namely, (3.6) for $i = 1$. Hence, let us continue to prove part (*ii*) under the assumption that $\Delta_1 \in (0,1)$ [recall (3.9)].

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 3.8 [in particular, (3.6) for $j = 1$ and $i = 1$] that

$$\frac{|d_{k+\hat{k},1,1}|}{|d_{k,1,1}|} \le \Delta_1^{\hat{k}} \text{ for any } \hat{k} \in \mathbb{N}. \tag{3.10}$$

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

$$\left\lceil \frac{\log \epsilon_1}{\log \Delta_1} \right\rceil \log \Delta_1 \le \left(\frac{\log \epsilon_1}{\log \Delta_1}\right) \log \Delta_1 = \log(\epsilon_1). \tag{3.11}$$

Since $\log(\cdot)$ is nondecreasing, the inequalities yielded by (3.11) combined with (3.10) and (3.6) 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 $\hat{j} \in [m]$. This is the reason for the addition of 1 to $\lceil \log \epsilon_1 / \log \Delta_1 \rceil$ in the general conclusion. □

One may conclude from Lemma 3.9 and (2.10) that, for any $(k,j) \in \mathbb{N} \times [m]$ and $\epsilon_1 \in (0,1)$, one has

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

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 (2.10) 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 holds for any $j \in [m]$ and $\hat{j} = j$.) For the lemma, we define the following constants that depend only on p, the spectrum of A (which, in particular, yields the bounds and definitions in Lemma 3.8) and the scalar constant $\rho \geq 1$ from Assumption 3.4:

$$\hat{\delta}_p := \left(1 + \hat{\delta}_{1,p}^2 + \hat{\delta}_{1,p}^2\hat{\delta}_{2,p}^2 + \cdots + \prod_{j=1}^{m-1}\hat{\delta}_{j,p}^2\right) \in [1,\infty), \tag{3.12a}$$

$$\hat{\Delta}_{p+1} := \max\left\{\frac{1}{3}, 1 - \frac{\lambda_{p+1}}{\lambda_n}\right\}^m \in (0,1), \tag{3.12b}$$

$$\text{and } \ \hat{K}_p := \left\lceil \frac{\log\left(2\hat{\delta}_p\rho\epsilon_p\Delta_{p+1}^{-(K_p+1)}\right)}{\log\hat{\Delta}_{p+1}} \right\rceil. \tag{3.12c}$$

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 with

$$\sum_{i=1}^{p} d_{k+\hat{k},1,i}^2 \leq \epsilon_p^2\|g_{k,1}\|^2 \ \text{ for all } \ \hat{k} \geq K_p, \tag{3.13}$$

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

$$d_{k+K_{p+1},1,p+1}^2 \leq 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2; \tag{3.14}$$

(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

$$d_{k+\hat{k}_0,1,p+1}^2 \leq 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2. \tag{3.15}$$

Proof. By Lemma 3.8 [in particular, (3.6) with $j = 1$ and $i = p+1$] and (3.4), it follows that

$$d_{k+\hat{k},1,p+1}^2 \leq \left(\Delta_{p+1}^{\hat{k}} d_{k,1,p+1}\right)^2 = \Delta_{p+1}^{2\hat{k}} d_{k,1,p+1}^2 \leq \Delta_{p+1}^{2\hat{k}}\|g_{k,1}\|^2 \ \text{ for all } \ \hat{k} \in \mathbb{N}. \tag{3.16}$$

If $\Delta_{p+1} \in [0, 1)$, then (3.16) immediately implies the existence of K_{p+1}, dependent only on ϵ_p, ρ and the spectrum of A such that (3.14) 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

$$d^2_{k+\hat{k},1,p+1} > 4\hat{\delta}_p^2 \rho^2 \epsilon_p^2 \|g_{k,1}\|^2 \;\text{ for all }\; \hat{k} \in \{K_p, \ldots, K_{p+1} - 1\}, \tag{3.17}$$

then (3.15) holds at the beginning of the next cycle (i.e., when $\hat{k}_0 = K_{p+1}$). From Lemma 3.7, Lemma 3.8 [in particular, (3.7)], (3.13) and (3.17), 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

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

Along with Lemma 3.7, this implies that

$$\theta_{k+\hat{k},j} = \sum_{i=1}^{n} \lambda_i c^2_{k+\hat{k},j,i} \geq \tfrac{3}{4}\lambda_{p+1} \;\text{ for all }\; \hat{k} \in \{K_p, \ldots, K_{p+1} - 1\} \;\text{ and }\; j \in [m]. \tag{3.18}$$

Together with Lemma 2.3 [see (2.11)] and $\alpha_{k+\hat{k}+1,j} = \theta^{-1}_{k+\hat{k},j}$ for all $j \in [m]$, the bound (3.18) implies

$$\begin{aligned}
d^2_{k+\hat{k}+2,1,p+1} &= \left(\prod_{j=1}^{m} (1 - \alpha_{k+\hat{k}+1,j}\lambda_{p+1})^2\right) d^2_{k+\hat{k}+1,1,p+1} \\
&\leq \hat{\Delta}^2_{p+1} d^2_{k+\hat{k}+1,1,p+1} \;\text{ for all }\; \hat{k} \in \{K_p, \ldots, K_{p+1} - 1\}.
\end{aligned} \tag{3.19}$$

Applying this bound recursively, it follows with $K_{p+1} = K_p + \hat{K}_p + 1$ and (3.16) for $\hat{k} = K_{p+1}$ that

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

where the last inequality follows by the definition of $\hat{K}_p$ in (3.12c). □

We have shown that small squared weights in (2.10) 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 3.11 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 (3.13) 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 3.10,

$$\sum_{i=1}^{p+1} d_{k+\hat{k},1,i}^2 \leq \epsilon_{p+1}^2 \|g_{k,1}\|^2 \text{ for all } \hat{k} \geq K_{p+1}. \tag{3.20}$$

Proof. For the same reasons as in the proof of Lemma 3.10, 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 (3.12). By Lemma 3.10, there exists $\hat{k}_0 \in \{K_p, \ldots, K_{p+1}\}$ such that

$$d_{k+\hat{k},1,p+1}^2 \leq 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \text{ when } \hat{k} = \hat{k}_0. \tag{3.21}$$

If the inequality in (3.21) holds for all $\hat{k} \geq \hat{k}_0$, then (3.20) 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

$$d_{k+\hat{k},1,p+1}^2 \leq 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \text{ for all } \hat{k}_0 \leq \hat{k} \leq \hat{k}_1, \tag{3.22}$$

but

$$d_{k+\hat{k}_1+1,1,p+1}^2 > 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2. \tag{3.23}$$

As in the arguments that lead to (3.19) in the proof of Lemma 3.10, combining (3.13) and (3.23) 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

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

Since $\hat{\Delta}_{p+1} \in (0,1)$, this fact and (3.23) imply the existence of $\hat{k}_2 \in \mathbb{N}$ such that

$$d_{k+\hat{k}+1,1,p+1}^2 > 4\hat{\delta}_p^2\rho^2\epsilon_p^2\|g_{k,1}\|^2 \text{ for all } \hat{k}_1 \leq \hat{k} \leq \hat{k}_2 - 2, \tag{3.24}$$

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,

$$d_{k+\hat{k}+3,1,p+1}^2 \leq \hat{\Delta}_{p+1}^2 d_{k+\hat{k}+2,1,p+1}^2 \text{ for all } \hat{k}_1 \leq \hat{k} \leq \hat{k}_2 - 2. \tag{3.25}$$

Moreover, by Lemma 3.8 [in particular, (3.6)] and (3.22), it follows that

$$d^2_{k+\hat{k}_1+1,1,p+1} \le \Delta^2_{p+1} d^2_{k+\hat{k}_1,1,p+1} \le 4\Delta^2_{p+1}\hat{\delta}^2_p \rho^2 \epsilon^2_p \|g_{k,1}\|^2 \tag{3.26a}$$

$$\text{and} \quad d^2_{k+\hat{k}_1+2,1,p+1} \le 4\Delta^4_{p+1}\hat{\delta}^2_p \rho^2 \epsilon^2_p \|g_{k,1}\|^2. \tag{3.26b}$$

Combining (3.25) and (3.26b), it follows that

$$d^2_{k+\hat{k}+3,1,p+1} \le 4\hat{\Delta}^2_{p+1}\Delta^4_{p+1}\hat{\delta}^2_p \rho^2 \epsilon^2_p \|g_{k,1}\|^2 \text{ for all } \hat{k}_1 \le \hat{k} \le \hat{k}_2 - 2.$$

Overall, since (3.12b) ensures $\hat{\Delta}_{p+1} \in (0, 1)$, we have shown that

$$d^2_{k+\hat{k},1,p+1} \le 4\Delta^4_{p+1}\hat{\delta}^2_p \rho^2 \epsilon^2_p \|g_{k,1}\|^2 \text{ for all } \hat{k} \in \{\hat{k}_0, \ldots, \hat{k}_2\}. \tag{3.27}$$

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

The following lemma is a generalization of Lemma 3.2 for any $m \in [n]$. Our proof is similar to that of Lemma 2.4 in Dai & Liao (2002). We provide it in full for completeness.

LEMMA 3.12 There exists $K \in \mathbb{N}$ dependent only on the spectrum of A such that

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

Proof. By Lemma 3.11, if for some $(\epsilon_p, K_p) \in (0, \frac{1}{2\hat{\delta}_p\rho}) \times \mathbb{N}$ independent of k, one finds

$$\sum_{i=1}^{p} d^2_{k+\hat{k},1,i} \le \epsilon^2_p \|g_{k,1}\|^2 \text{ for all } \hat{k} \ge K_p, \tag{3.28}$$

then for $\epsilon^2_{p+1} := (1 + 4\max\{1, \Delta^4_{p+1}\}\hat{\delta}^2_p\rho^2)\epsilon^2_p$ and some $K_{p+1} \ge K_p$ independent of k, one finds

$$\sum_{i=1}^{p+1} d^2_{k+\hat{k},1,i} \le \epsilon^2_{p+1} \|g_{k,1}\|^2 \text{ for all } \hat{k} \ge K_{p+1}. \tag{3.29}$$

Since Lemma 3.9 implies that for any $\epsilon_1 \in (0, 1)$, one can find K_1 independent of k such that (3.28) holds with $p = 1$, it follows that, independent of k, there exists a sufficiently small $\epsilon_1 \in (0, 1)$ such that

$$\epsilon^2_1 \le \cdots \le \epsilon^2_n \le \tfrac{1}{4}.$$

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

$$\|g_{k+\hat{k},1}\|^2 = \sum_{i=1}^{n} d^2_{k+\hat{k},1,i} \le \tfrac{1}{4}\|g_{k,1}\|^2 \text{ for all } \hat{k} \ge K,$$

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 Dai & Liao (2002). We prove it in full for completeness.

THEOREM 3.13 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 3.8) that $\{\|g_{k,1}\|\}$ vanishes *Q*-linearly. Hence, let us continue assuming that $\Delta \geq 1$. By Lemma 3.12, there exists $K \in \mathbb{N}$ dependent only on the spectrum of A such that,

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

Applying this result recursively, it follows that

$$\|g_{1+Kl,1}\| \leq (\tfrac{1}{2})^{l}\|g_{1,1}\| \text{ for all } l \in \mathbb{N}. \tag{3.30}$$

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, (3.8) and (3.30), 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

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

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 Fletcher (2012). 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 article. 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 3.8. 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

TABLE 1 *Spectra of A for five test problems along with outer and (total) inner iteration counts required by Algorithm LMSD and maximum value of the ratio $\|R_k^{-1}\|/(\|g_{k,1}\|^{-1})$ observed during the run of Algorithm LMSD. For each spectrum, a set of eigenvalues in an interval indicates that the eigenvalues are evenly distributed within that interval*

		$m = 1$			$m = 5$		
Problem	Spectrum	k	j	ρ	k	j	ρ
1	$\{\lambda_1, \ldots, \lambda_{100}\} \subset [1, 1.9]$	13	13	1	3	14	$\sim 6 \times 10^3$
2	$\{\lambda_1, \ldots, \lambda_{100}\} \subset [1, 100]$	124	124	1	23	114	$\sim 1 \times 10^4$
3	$\{\lambda_1, \ldots, \lambda_{20}\} \subset [1, 2]$ $\{\lambda_{21}, \ldots, \lambda_{40}\} \subset [25, 26]$ $\{\lambda_{41}, \ldots, \lambda_{60}\} \subset [50, 51]$ $\{\lambda_{61}, \ldots, \lambda_{80}\} \subset [75, 76]$ $\{\lambda_{81}, \ldots, \lambda_{100}\} \subset [99, 100]$	112	112	1	16	79	$\sim 2 \times 10^5$
4	$\{\lambda_1, \ldots, \lambda_{99}\} \subset [1, 2]$ $\lambda_{100} = 100$	26	26	1	4	20	$\sim 2 \times 10^{16}$
5	$\lambda_1 = 1$ $\{\lambda_2, \ldots, \lambda_{100}\} \subset [99, 100]$	16	16	1	5	25	$\sim 2 \times 10^{10}$

$[\lambda_1, \lambda_n]$; 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.

Table 1 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 step sizes were generated randomly from a uniform distribution over the interval $[\lambda_{100}^{-1}, \lambda_1^{-1}]$. One finds that the algorithm terminated 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 3.6, 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 (2.10) will vanish quickly. On the other hand, for the problems for which the eigenvalues are more evenly spread in [1, 100], the algorithm required relatively more outer iterations, though still not an excessively large number relative to n. For these problems, the performance was mostly better for $m = 5$ vs. $m = 1$, in terms of both outer and (total) inner iterations.

In Table 1, we also provide the maximum over k of the ratio $\|R_k^{-1}\|/(\|g_{k,1}\|^{-1})$ (indicated by the column header 'ρ') observed during the run of the algorithm for each test problem and each m. The purpose of this is to confirm that Assumption 3.4 indeed held in our numerical experiments, but also to demonstrate for what value of ρ the assumption holds. As explained following Assumption 3.4, for $m = 1$, the ratio was always equal to 1. As for $m = 5$, on the other hand, the ratio was sometimes quite large, though it is worthwhile to remark that the ratio was typically much smaller than this maximum value. We did not observe any predictable behavior about when this maximum value was observed; sometimes it occurred early in the run, while sometimes it occurred toward the end. Overall, the evolution of this ratio depends on the initial point and path followed by the algorithm to the minimizer.

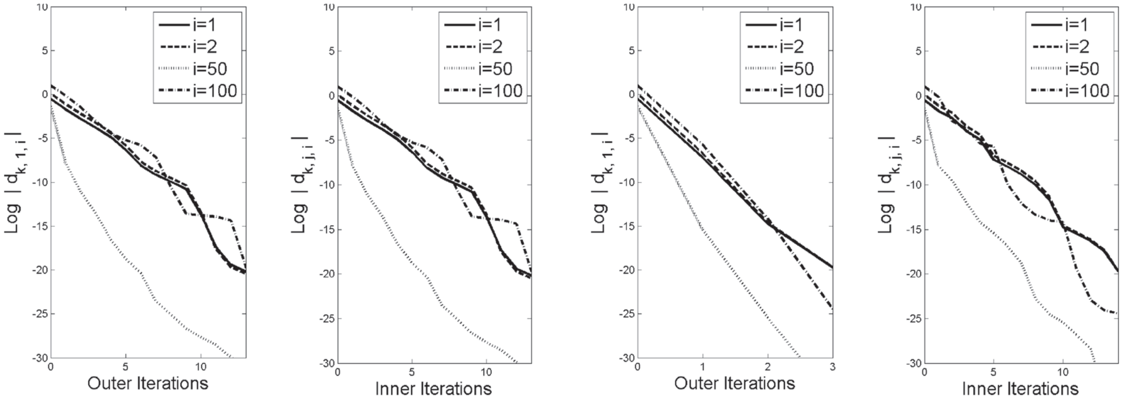

FIG. 1. Weights in (2.10) for problem 1 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

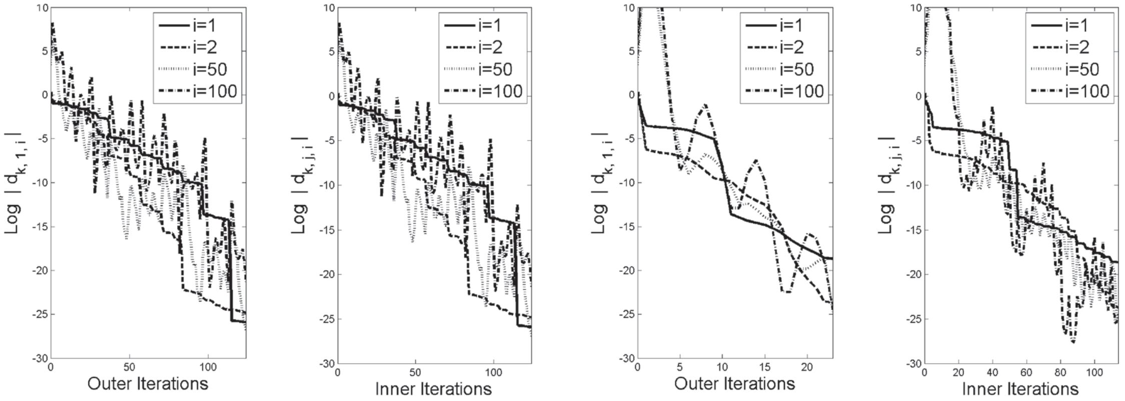

FIG. 2. Weights in (2.10) for problem 2 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

As seen in our analysis (inspired by Raydan, 1993; Dai & Liao, 2002; Fletcher, 2012), a more refined look into the behavior of the algorithm is obtained by observing the step-by-step magnitudes of the weights in (2.10) for the generated gradients. Hence, for each of the test problems, we plot in Figs 1– 5 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 iterations (including inner ones), 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 figures show that the magnitudes of the weights corresponding to $i = 1$ always decrease monotonically, as proved in Lemma 3.9. 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.

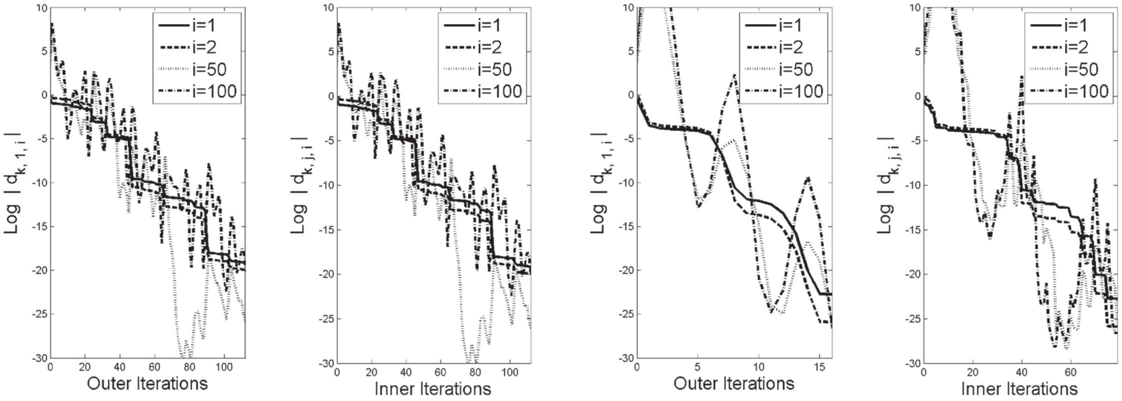

FIG. 3. Weights in (2.10) for problem 3 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

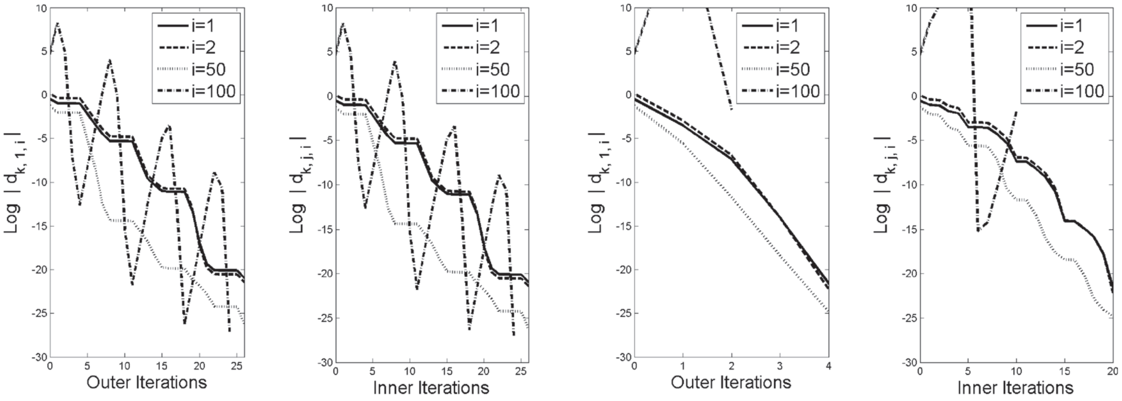

FIG. 4. Weights in (2.10) for problem 4 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

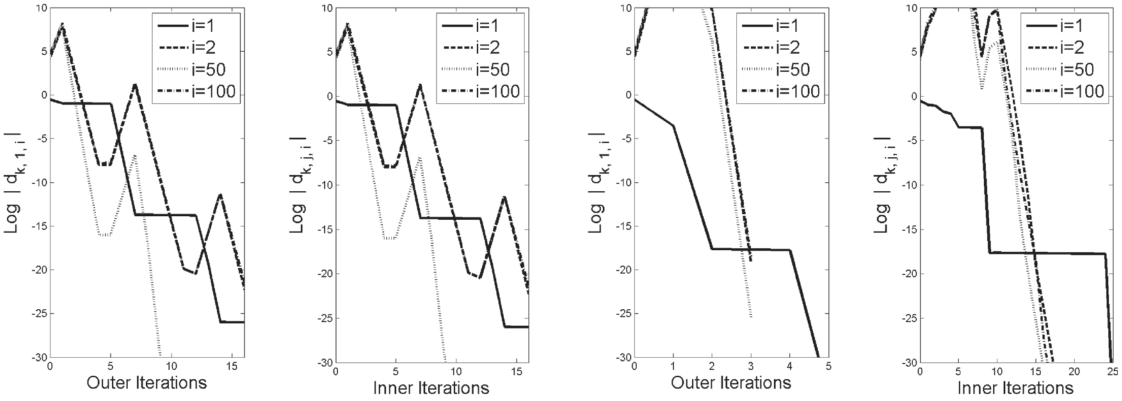

FIG. 5. Weights in (2.10) for problem 5 with history length $m = 1$ (left two plots) and $m = 5$ (right two plots).

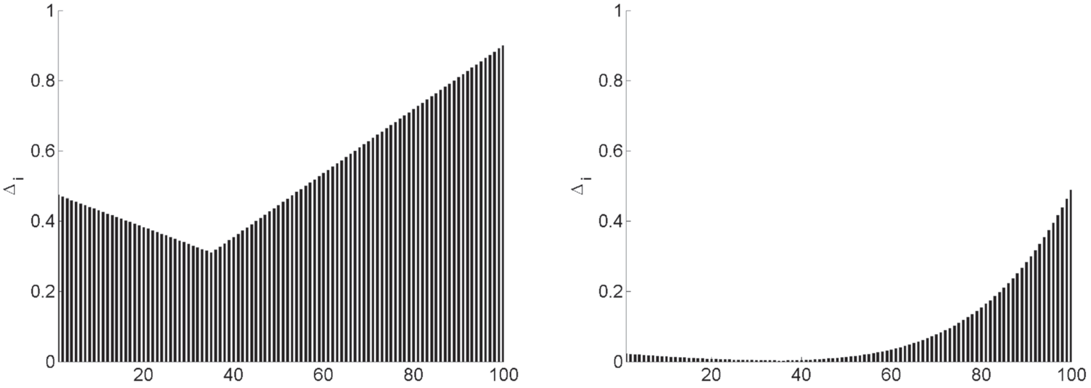

FIG. 6. Constants in (3.6) for problem 1 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

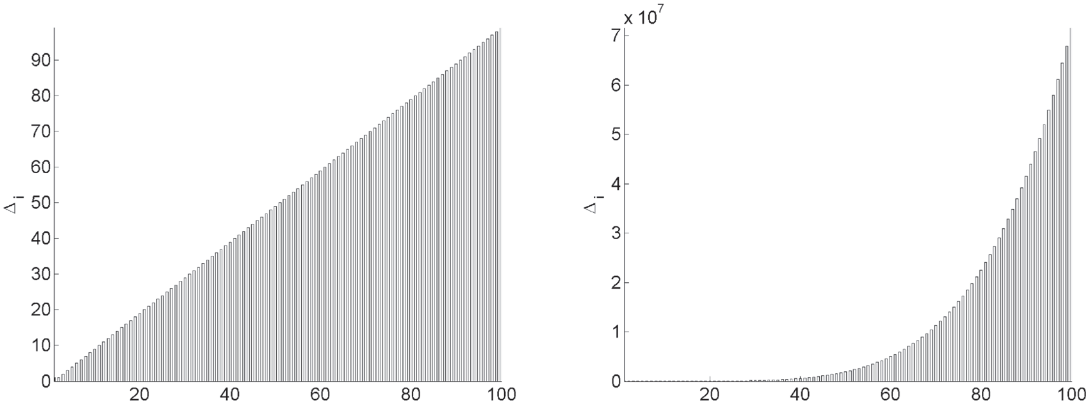

FIG. 7. Constants in (3.6) for problem 2 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

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 (3.6), these constants bound the increase that a particular weight in (2.10) might experience from one point in a cycle to the same point in the subsequent cycle. For illustration, we plot in Figs 6–10 these constants. Values less than 1 are indicated by a black bar, while values greater than or equal to 1 are indicated by a gray bar. Note that, in Fig. 9, all values are small for both history lengths except Δ_{100}. In Fig. 10, Δ_1 is less than 1 in both figures, but the remaining constants are large for $m = 1$ while being small for $m = 5$.

5. Conclusion

We have shown that the LMSD method proposed by Fletcher (2012) possesses an R-linear rate of convergence for any history length $m \in [n]$, when it is employed to minimize a strongly convex quadratic function. Our analysis effectively extends that in Dai & Liao (2002), which covers only the $m = 1$ case. We have also provided the results of numerical experiments to demonstrate that the behavior of the algorithm reflects certain properties revealed by the theoretical analysis.

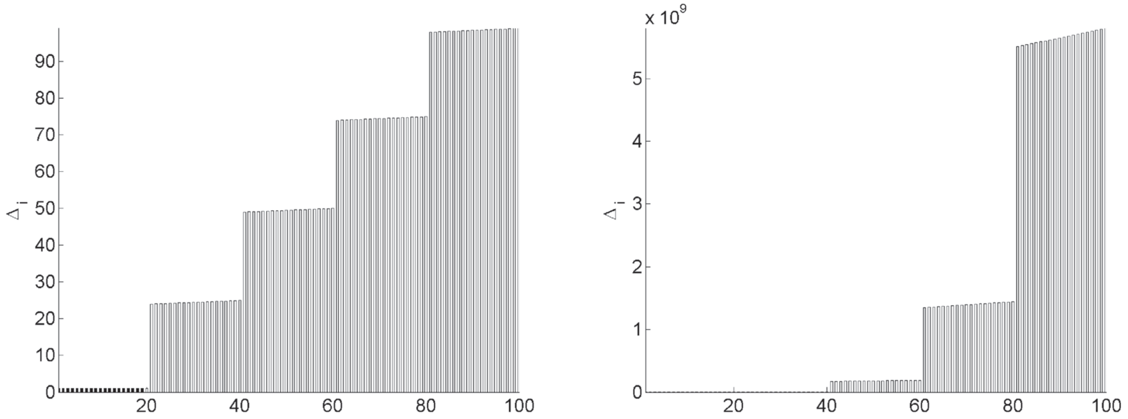

FIG. 8. Constants in (3.6) for problem 3 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

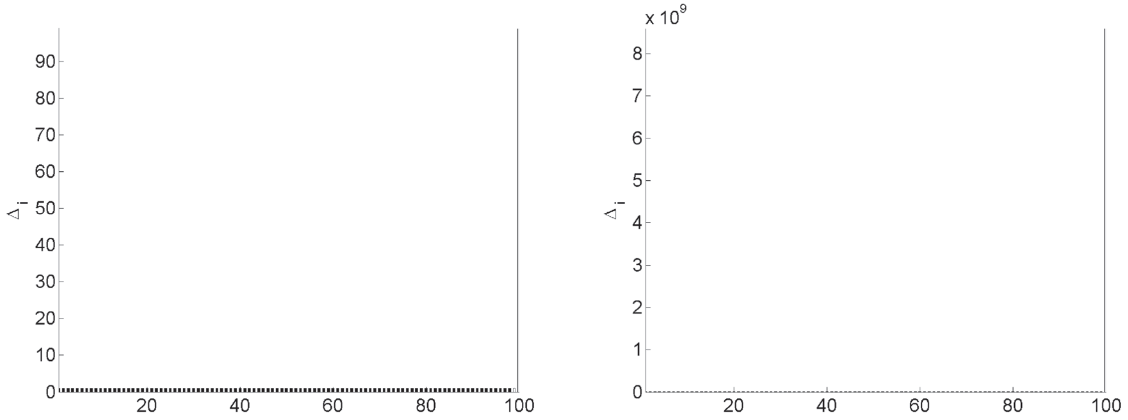

FIG. 9. Constants in (3.6) for problem 4 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

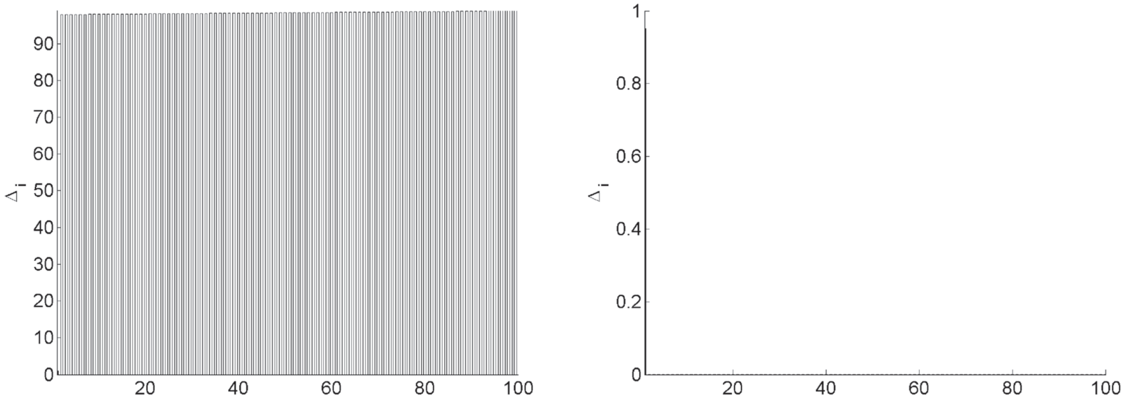

FIG. 10. Constants in (3.6) for problem 5 with history length $m = 1$ (left plot) and $m = 5$ (right plot).

One might wonder whether the convergence rate of LMSD is the same when Ritz values are replaced by harmonic Ritz values; see Fletcher (2012, Section 7). We answer this in the affirmative in the appendix.

Acknowledgements

The authors would like to thank the Associate Editor for handling the submission of this paper as well as the two anonymous referees whose comments and suggestions helped to improve it. They would also like to thank Prof. Yu-Hong Dai, whose presentation at the Fifth International Conference on Continuous Optimization was an inspiration to complete this work.

Funding

U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics, Early Career Research Program [contract number DE–SC0010615]; U.S. National Science Foundation [grant nos DMS–1319356 and CCF-1618717].

References

Barzilai, J. & Borwein, J. M. (1988) Two-point step size gradient methods. *IMA J. Numer. Anal.*, **8**, 141–148.

Beattie, C. (1998) Harmonic Ritz and Lehmann bounds. *Electron. Trans. Numer. Anal.*, **7**, 18–39.

Curtis, F. E. & Guo, W. (2016) Handling nonpositive curvature in a limited memory steepest descent method. *IMA J. Numer. Anal.*, **36**, 717–742.

Dai, Y.-H. & Liao, L.-Z. (2002) *R*-linear convergence of the Barzilai and Borwein gradient method. *IMA J. Numer. Anal.*, **22**, 1–10.

Fletcher, R. (2012) A limited memory steepest descent method. *Math. Program.*, **135**, 413–436.

Lai, Y. L. (1981) Some properties of the steepest descent method. *Acta Mathematicae Applicatae Sinica*, **4**, 106–116.

Paige, C. C., Parlett, B. N. & van der Vorst, H. (1995) Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numer. Linear Algebra Appl.*, **2**, 115–133.

Parlett, B. N. (1998) *The Symmetric Eigenvalue Problem*. Classics in Applied Mathematics. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Raydan, M. (1993) On the Barzilai and Borwein choice of steplength for the gradient method. *IMA J. Numer. Anal.*, **13**, 321–326.

Appendix. LMSD with harmonic Ritz values

As explained in Fletcher (2012, Section 7), an alternative limited memory steepest descent method (LMSD) is one that replaces Ritz values of A with harmonic Ritz values (see also Curtis & Guo, 2016). In the case of $m = 1$, this reduces to replacing the 'first' with the 'second' BB step size formula; see in Barzilai & Borwein (1988, (5)–(6)). In this appendix, we briefly describe the differences in the computations involved in this alternative approach, then argue that the resulting algorithm is also R-linearly convergent. In fact, much of the analysis in Section 3.2 remains true for this alternative algorithm, so here we only highlight the minor differences.

First, let us briefly review the differences in the computations involved in this alternative algorithm. For this, we follow the description in Fletcher (2012). Recalling that $T_k = Q_k^T A Q_k$, the Ritz values used in Algorithm LMSD can be viewed as being determined by the eigensystem $(Q_k^T A Q_k)V = (Q_k^T Q_k)V\Theta$, i.e., the solution of this system has $\Theta = \text{diag}(\theta_{k,1}, \dots, \theta_{k,m})$. Including another instance of A on both sides of this system, one obtains the generalized eigensystem $(Q_k^T A^2 Q_k)V = (Q_k^T A Q_k)V\Theta$, the eigenvalues of which are referred to as harmonic Ritz values of A (see Paige *et al.*, 1995). Defining $P_k := Q_k^T A^2 Q_k$, the eigenvalues are those of $(P_k^{-1}T_k)^{-1}$, which we shall denote as $\{\mu_{k,j}\}_{j\in[m]} \subset \mathbb{R}_{++}$ in decreasing order. The alternative LMSD method is simply Algorithm LMSD with $\{\mu_{k,j}\}_{j\in[m]}$ in place of $\{\theta_{k,j}\}_{j\in[m]}$. As explained in Fletcher (2012), the matrix P_k, like T_k, can be computed with relative little computation and storage, and without explicit access to A. In particular, if G_k has linearly independent columns, one can compute upper triangular $R_k \in \mathbb{R}^{m\times m}$, $r_k \in \mathbb{R}^m$ and $\xi_k \in \mathbb{R}$ from the Cholesky factorization

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

then, with J_k again from (2.5), compute

$$P_k \leftarrow R_k^{-T} J_k^T \begin{bmatrix} R_k & r_k \\ & \xi_k \end{bmatrix}^T \begin{bmatrix} R_k & r_k \\ & \xi_k \end{bmatrix} J_k R_k^{-1}. \tag{A.2}$$

One also finds that $P_k = T_k^T T_k + \zeta_k \zeta_k^T$, where $\zeta_k^T = \begin{bmatrix} 0 & \xi_k \end{bmatrix} J_k R_k^{-1}$ (see Curtis & Guo, 2016).

Let us now argue that this alternative LMSD method is R-linearly convergent. For this, we first show that the harmonic Ritz values satisfy the same property as shown for the Ritz values in Lemma 3.6.

Lemma A.1 Given T_k from (2.8) and P_k from (A.2), the eigenvalues $\{\mu_{k,j}\}_{j\in[m]}$ of $(P_k^{-1}T_k)^{-1}$ satisfy

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

Proof. One can apply, e.g., Theorem 2.1 from Beattie (1998) with ‘K’$= A$, ‘M’$= I$, and ‘P’$= Q_k$, the proof of which follows from min–max characterizations of the eigenvalues. □

Given Lemma A.1, one can verify that the results shown in Lemmas 3.8 and 3.9 also hold for our alternative LMSD method. The result in Lemma 3.10 remains true as well, though the argument for this requires a slight addition to the proof. First, we need the following known property that the Ritz and harmonic Ritz values are interlaced (e.g., see Curtis & Guo, 2016, Theorem 3.3).

Lemma A.2 Given T_k from (2.8) and P_k from (A.2), the eigenvalues $\{\theta_{k,j}\}_{j\in[m]}$ of T_k and the eigenvalues $\{\mu_{k,j}\}_{j\in[m]}$ of $(P_k^{-1}T_k)^{-1}$ are interlaced in the sense that

$$\mu_{k,1} \geq \theta_{k,1} \geq \mu_{k,2} \geq \theta_{k,2} \geq \cdots \geq \mu_{k,m} \geq \theta_{k,m} > 0.$$

Using this result, let us now argue that Lemma 3.10 remains true. Indeed, our previous proof remains valid through the statement of (3.18). Then, combining (3.18) with Lemma A.2, one may conclude that

$$\mu_{k+\hat{k},j} \geq \theta_{k+\hat{k},j} \geq \tfrac{3}{4}\lambda_{p+1} \text{ for all } \hat{k} \in \{K_p, \ldots, K_{p+1}-1\} \text{ and } j \in [m].$$

The remainder of the proof then follows as before with $\alpha_{k+\hat{k}+1,j} = \mu_{k+\hat{k},j}^{-1}$ for all $j \in [m]$. A similar modification is needed in the proof of Lemma 3.11 since it employs a similar argument as in the proof of Lemma 3.10. This way, one can verify that Lemma 3.11 remains true for the alternative LMSD method. Finally, as for Lemma 3.12 and Theorem 3.13, our proofs follow as before without any modifications.