IMA Journal of Numerical Analysis (2020) **00**, 1–27
doi:10.1093/imanum/drz052

Analysis of limited-memory BFGS on a class of nonsmooth convex functions

AZAM ASL* AND MICHAEL L. OVERTON
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 USA
*Corresponding author: aa2821@nyu.edu

[Received on 30 September 2018; revised on 18 September 2019]

The limited-memory BFGS (Broyden-Fletcher-Goldfarb-Shanno) method is widely used for large-scale unconstrained optimization, but its behavior on nonsmooth problems has received little attention. L-BFGS (limited memory BFGS) can be used with or without 'scaling'; the use of scaling is normally recommended. A simple special case, when just one BFGS update is stored and used at every iteration, is sometimes also known as memoryless BFGS. We analyze memoryless BFGS with scaling, using any Armijo–Wolfe line search, on the function $f(x) = a|x^{(1)}| + \sum_{i=2}^{n} x^{(i)}$, initiated at any point x_0 with $x_0^{(1)} \neq 0$. We show that if $a \geq 2\sqrt{n-1}$, the absolute value of the normalized search direction generated by this method converges to a constant vector, and if, in addition, a is larger than a quantity that depends on the Armijo parameter, then the iterates converge to a nonoptimal point $\bar{x}$ with $\bar{x}^{(1)} = 0$, although f is unbounded below. As we showed in previous work, the gradient method with any Armijo–Wolfe line search also fails on the same function if $a \geq \sqrt{n-1}$ and a is larger than another quantity depending on the Armijo parameter, but scaled memoryless BFGS fails under a *weaker* condition relating a to the Armijo parameter than that implying failure of the gradient method. Furthermore, in sharp contrast to the gradient method, if a specific standard Armijo–Wolfe bracketing line search is used, scaled memoryless BFGS fails when $a \geq 2\sqrt{n-1}$ *regardless* of the Armijo parameter. Finally, numerical experiments indicate that the results may extend to scaled L-BFGS with any fixed number of updates m, and to more general piecewise linear functions.

Keywords: nonsmooth optimization; BFGS.

1. Introduction

The limited-memory BFGS (Broyden-Fletcher-Goldfarb-Shanno) method is widely used for large-scale unconstrained optimization, but its behavior on nonsmooth problems has received little attention. In this paper we give the first analysis of an instance of the method, sometimes known as memoryless BFGS with scaling, on a specific class of nonsmooth convex problems, showing that under given conditions the method generates iterates whose function values are bounded below, although the function itself is unbounded below.

The 'full' BFGS method (Nocedal & Wright, 2006, Sec. 6.1), independently derived by Broyden, Fletcher, Goldfarb and Shanno in 1970, is remarkably effective for unconstrained optimization, but even when the minimization objective $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be twice continuously differentiable and convex, with bounded level sets, the analysis of the method is nontrivial. Powell (1976) gave the first convergence analysis for full BFGS using an Armijo–Wolfe line search for this class of functions, establishing convergence to the minimal function value. In the smooth, nonconvex case it is generally accepted that the method is very reliable for finding stationary points (usually local minimizers), although pathological counterexamples exist (Dai, 2002; Mascarenhas, 2004).

At first glance, it might appear that, since BFGS uses gradient differences to approximate information about the Hessian of f, the use of BFGS for nonsmooth optimization makes little sense. Firstly, because at minimizers where f is not differentiable, neither the gradient nor the Hessian exists; and secondly, even at other points where f is twice differentiable, the Hessian might appear to be meaningless. For example, for a piecewise linear function such as studied in this paper, the Hessian is zero everywhere that it is defined. However, the way to make sense of the applicability of BFGS to a nonsmooth function is to consider its approximation by a very ill-conditioned smooth function. For example, the function $f(x) = \|x\|_2$ can be arbitrarily well approximated by the smooth function $f(x) = \sqrt{\|x\|_2^2 + \epsilon^2}$, where $\epsilon > 0$. As $\epsilon \downarrow 0$, the approximation becomes arbitrarily good—but also arbitrarily ill-conditioned. For any *fixed* $\epsilon > 0$, the BFGS convergence theory applies. As $\epsilon \downarrow 0$, it is not at all clear what impact the property of good approximation via badly conditioned functions has on the convergence theory, which, of course, does not apply when $\epsilon = 0$. Nonetheless, even for $\epsilon = 0$, the method remains well defined, as the gradient is defined everywhere except at the minimizer (the origin). In fact, it was established recently by Guo & Lewis (2018) that Powell's result for smooth functions mentioned above can be extended, in a nontrivial way, to show that the iterates generated by BFGS with an Armijo–Wolfe line search, when applied to $f(x) = \|x\|_2$, converge to the origin. Even the case $n = 1$, where f is the absolute value function, is surprisingly complex; it turns out that in this case the sequence of iterates is defined by a certain binary expansion of the starting point (Lewis & Overton, 2013). However, in this simple example it is easy to see intuitively *why* BFGS works well. The line search ensures that the iterates oscillate back and forth across the origin, giving a gradient difference equal to 2 at every iteration. As the iterates converge to the origin, the result is that the 'inverse Hessian approximation' generated by BFGS converges to zero, resulting in quasi-Newton steps that also converge to zero. An important consequence is that the line search never requires many function evaluations. In contrast, when gradient descent with the same line search is applied to the absolute value function, the iterates converge to the origin, but each line search requires a number of function evaluations that increases with the iteration number.

More generally, if f is locally Lipschitz, BFGS is still typically well defined, because such functions are differentiable almost everywhere by Rademacher's theorem (Clarke, 1990), and hence f is differentiable at a randomly generated point with probability one. Furthermore, substantial computational experience (Lewis & Overton, 2013) shows that even when f is nonsmooth, the method is remarkably reliable for finding Clarke stationary points (again, typically local minimizers), and furthermore, this property extends in a certain sense to constrained problems (Curtis *et al.*, 2017). Indeed, no nonpathological counterexamples showing convergence to nonstationary values, meaning in particular examples where the starting point is not predetermined but generated randomly, are known. The superlinear convergence rate that holds generically for smooth functions is not attained in the nonsmooth case; instead, full BFGS is observed to converge linearly, in a sense described in Lewis & Overton (2013), on nonsmooth functions. Furthermore, in general one does not observe the inverse Hessian approximation converging to zero; instead, what seems to be typical is that *some* of its eigenvalues converge to zero, with corresponding eigenvectors identifying directions along which f is nonsmooth at the minimizer. See Lewis & Overton (2013, Sec. 6.2) for details.

The full BFGS method maintains and updates an approximation to the inverse (or a factorization) of the Hessian matrix $\nabla^2 f(x)$ at every iteration, defined by current known gradient difference information $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$ along with $s_{k-1} = x_k - x_{k-1}$. The use of the Wolfe condition in the line search, requiring an increase in the directional derivative of f along the descent direction generated by BFGS, ensures that the updated inverse Hessian approximation is positive definite. Since the update has rank two, the cost of full BFGS is $O(n^2)$ operations per iteration. While this was a great advance over the cost of Newton's method in the 1970s, already in the 1980s it was realized that the cost was too high for problems where n is large, and hence the limited-memory version, L-BFGS, became popular, and is widely used today (see Le *et al.*, 2011, Mokhtari & Ribeiro, 2015 and Lin *et al.*, 2016, for example). The standard version of L-BFGS was introduced by Liu & Nocedal (1989) and is also discussed in detail in Nocedal & Wright (2006, Sec. 7.2). Let $m \ll n$ be given. Instead of maintaining an approximation to the inverse Hessian, at the kth iteration a proxy for this matrix is implicitly defined by application of the most recent m BFGS updates (which are defined by saving y_j and s_j from the past m iterations) to a given sparse matrix H_k^0. One possible choice for H_k^0 is the identity matrix I, but a popular choice is to instead use *scaling*, defining

$$H_k^0 = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}} I. \tag{1}$$

Analysis of L-BFGS is more straightforward than analysis of full BFGS in the case that f is smooth and strongly convex, and is given in Liu & Nocedal (1989, Theorem 7.1), where linear convergence to minimizers is established, regardless of whether scaling is used or not. Furthermore, it is stated in Liu & Nocedal (1989) that scaling greatly accelerates L-BFGS, and this seems to be the currently accepted wisdom. However, we show in this paper that it is exactly the choice of scaling that may result in failure of L-BFGS on a specific class of nonsmooth functions. This situation is in sharp contrast to our experience with full BFGS on nonsmooth functions, where the same algorithm that is normally used for smooth functions works well also on nonsmooth functions.

We consider the convex function

$$f(x) = a|x^{(1)}| + \sum_{i=2}^{n} x^{(i)}, \tag{2}$$

where $a \geq \sqrt{n-1}$. Note that although f is unbounded below, it is bounded below along the line defined by the negative gradient direction from any point x with $x^{(1)} \neq 0$. In Asl & Overton (2019) we analyzed the gradient method with *any* Armijo–Wolfe line search applied to (2). We showed that if

$$a > \sqrt{\left(\frac{1}{c_1} - 1\right)(n-1)}, \tag{3}$$

where c_1 is the Armijo parameter, the gradient method, initiated at *any* point x_0 with $x_0^{(1)} \neq 0$, fails in the sense that it generates a sequence converging to a nonoptimal point $\bar{x}$ with $\bar{x}^{(1)} = 0$, although f is unbounded below. In the present paper, we analyze scaled L-BFGS with $m = 1$, i.e., with just one update—a method sometimes known as *memoryless* BFGS (Nocedal & Wright, 2006, p. 180)—applied to the function (2), and identify conditions under which the method converges to nonoptimal points (more details are given in the next paragraph). In contrast, it is known that when *full* BFGS is applied to the same function, eventually the method generates a search direction on which f is unbounded below (Xie & Waechter, 2017); see also Lewis & Zhang (2015). The specific choice of objective function f offers two advantages: one is its simplicity, but another is that there is little difficulty distinguishing in practice whether the method 'succeeds' or 'fails' from a given starting point; success is associated with a sequence of function values that is unbounded below, while convergence of the sequence to a finite value implies failure.

The paper is organized as follows. In Section 2, we define the scaled memoryless BFGS method, using any line search satisfying the Armijo and Wolfe conditions, and derive some properties of the method applied to the function f in (2), initiated at any point x_0 with $x_0^{(1)} \neq 0$. In Section 2.1, we show that if $a \geq \sqrt{3(n-1)}$, the algorithm is well defined in the sense that Armijo–Wolfe steplengths always exist, deferring the technical details to Appendix A. Then in Section 3, we give our main theoretical results. First, in Section 3.1, we show that if $a \geq 2\sqrt{n-1}$, in the limit the absolute value of the normalized search direction generated by the method converges to a constant vector, deferring the most technical parts of the proof to Appendix B. Then, in Section 3.2, we show that if a further satisfies a condition depending on the Armijo parameter, the method converges to a nonoptimal point $\bar{x}$ with $\bar{x}^{(1)} = 0$. Furthermore, this condition is *weaker* than the corresponding condition (3) for the gradient method. Then, in Section 3.3, we show that, if a specific standard Armijo–Wolfe bracketing line search is used, scaled memoryless BFGS converges to a nonoptimal point when $a \geq 2\sqrt{n-1}$ *regardless* of the Armijo parameter. This is in sharp contrast to the gradient method using the same line search, for which success or failure on the function f depends on the Armijo parameter. In Section 4 we present some numerical experiments which support our theoretical results, and which indicate that the results may extend to scaled L-BFGS with any fixed number of updates m, and to more general piecewise linear functions. We make some concluding remarks in Section 5.

2. The memoryless BFGS method

First let f denote any locally Lipschitz function mapping $\mathbb{R}^n$ to $\mathbb{R}$, and let $x_{k-1} \in \mathbb{R}^n$ denote the $(k-1)$th iterate of an optimization algorithm where f is differentiable at x_{k-1} with gradient $\nabla f(x_{k-1})$. Let $d_{k-1} \in \mathbb{R}^n$ denote a descent direction, i.e., satisfying $\nabla f(x_{k-1})^T d_{k-1} < 0$. Let parameters c_1 and c_2, known as the Armijo and Wolfe parameters, satisfy $0 < c_1 < c_2 < 1$. We say that the steplength t satisfies the Armijo condition at iteration $k-1$ if

$$f(x_{k-1} + td_{k-1}) \leq f(x_{k-1}) + c_1 t \nabla f(x_{k-1})^T d_{k-1} \tag{4}$$

and that it satisfies the Wolfe condition if

$$\nabla f(x_{k-1} + td_{k-1}) \text{ exists with } \nabla f(x_{k-1} + td_{k-1})^T d_{k-1} \geq c_2 \nabla f(x_{k-1})^T d_{k-1}. \tag{5}$$

It is known that if f is smooth or convex, and bounded below along the direction d_{k-1}, a point satisfying these conditions must exist (see Lewis & Overton, 2013, Theorem 4.5, for weaker conditions on f for which this holds). Note that as long as f is differentiable at the initial iterate, defining subsequent iterates by $x_k = x_{k-1} + t_{k-1} d_{k-1}$, where (5) holds for $t = t_{k-1}$, ensures that f is differentiable at x_k.

We are now ready to define the memoryless BFGS method (L-BFGS with $m = 1$), also known as L-BFGS-1, with scaling, i.e., with H_k^0 defined by (1). The algorithm is defined for any f, but its analysis will be specifically for (2).

Algorithm 1 (Memoryless BFGS with scaling), with input $\mathbf{x_0}$

$$d_0 = -\nabla f(x_0) \tag{6}$$

for $k = 1, 2, 3, \ldots,$ **define**

$t_{k-1} = t$ satisfying (4) and (5)

$$x_k = x_{k-1} + t_{k-1} d_{k-1} \tag{7}$$

$$s_{k-1} = x_k - x_{k-1} \tag{8}$$

$$y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}) \tag{9}$$

$$V_{k-1} = I - \frac{y_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} \tag{10}$$

$$H_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}} V_{k-1}^T V_{k-1} + \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T y_{k-1}} \tag{11}$$

$$d_k = -H_k \nabla f(x_k) \tag{12}$$

end

Let us adopt the convention that if no steplength t exists satisfying the Armijo and Wolfe conditions (4) and (5), the algorithm is terminated. Hence, for any smooth or convex function, termination implies that a direction d_{k-1} has been identified along which $f(x_{k-1} + td_{k-1})$ is unbounded below.

Now let us restrict our attention to the convex function f given in (2). The question we address in this paper is whether memoryless BFGS will succeed in identifying the fact that f is unbounded below, either because it generates a direction d for which no steplength t satisfying the Armijo and Wolfe conditions exists (in which case the algorithm terminates), or, alternatively, that it generates a sequence $\{x_k\}$ for which Armijo–Wolfe steps always exist, with $f(x_k) \downarrow -\infty$. If neither event takes place, $\{f(x_k)\}$ is bounded below, which is regarded as failure, since f is unbounded below.

For the function (2), requiring t_{k-1} to satisfy (5), regardless of the value of the Wolfe parameter $c_2 \in (0, 1)$, implies, via (7), the condition

$$\operatorname{sgn}(x_k^{(1)}) = -\operatorname{sgn}(x_{k-1}^{(1)}). \tag{13}$$

Via (8) we see that (13) is equivalent to the condition

$$|s_{k-1}^{(1)}| = |x_{k-1}^{(1)}| + |x_k^{(1)}|. \tag{14}$$

Without loss of generality, we assume that the initial point x_0 has a positive first component, i.e., $x_0^{(1)} > 0$, so that

$$\nabla f(x_k) = \begin{bmatrix} (-1)^k a \\ \mathbb{1} \end{bmatrix}, \tag{15}$$

where $\mathbb{1} \in \mathbb{R}^{n-1}$ is the column vector of all ones. Via (13) and (15), (9) is simply

$$y_{k-1} = \begin{bmatrix} (-1)^k 2a \\ \mathbb{0} \end{bmatrix}, \tag{16}$$

where $\mathbb{0} \in \mathbb{R}^{n-1}$ is the column vector of all zeros. Note that from (7) and (8) it is immediate that for any $k \geq 1$

$$s_{k-1} = t_{k-1} d_{k-1}. \tag{17}$$

For $i = 2, \ldots, n$, let

$$\theta_{k-1}^{(i)} = \arctan\left(\frac{d_{k-1}^{(i)}}{d_{k-1}^{(1)}}\right),$$

with $\theta_{k-1}^{(i)} \in [-\pi/2, \pi/2]$. Note that $|\theta_{k-1}^{(i)}|$ is the acute angle between d_{k-1} and the $x^{(1)}$ axis when it is projected onto the $(x^{(1)}, x^{(i)})$ plane. From (6) and (15) we have

$$\frac{1}{a} = \tan\theta_0^{(2)} = \tan\theta_0^{(3)} = \ldots = \tan\theta_0^{(n)}. \tag{18}$$

The assumption of the initial inverse Hessian approximation being a multiple of the identity is embedded in the definition (11), and therefore we know that d_{k-1} (and consequently s_{k-1}) is in the subspace spanned by the two gradients in (15) (see Gill & Leonard, 2003, Lemma 2.1). Since both gradients are symmetric w.r.t. the components $x^{(2)}, \ldots, x^{(n)}$, it follows that d_{k-1} has the same property. The same symmetry holds in the definition of the objective function (2). Since (18) holds, we conclude inductively that, for $k > 1$, $\tan\theta_{k-1}^{(2)} = \tan\theta_{k-1}^{(3)} = \ldots = \tan\theta_{k-1}^{(n)}$. So, let us simply write

$$b_{k-1} = \tan\theta_{k-1} = \frac{d_{k-1}^{(i)}}{d_{k-1}^{(1)}} = \frac{s_{k-1}^{(i)}}{s_{k-1}^{(1)}}, \text{for all } i = 2, \ldots, n. \tag{19}$$

From (16) we have

$$s_{k-1}^T y_{k-1} = (-1)^k 2a s_{k-1}^{(1)}, \tag{20}$$

so we can rewrite V_{k-1} in (10) in terms of b_{k-1} as

$$V_{k-1} = \left[\begin{array}{c|c} 0 & -b_{k-1}\mathbb{1}^T \\ \mathbb{0} & I_{n-1} \end{array}\right]. \tag{21}$$

This leads us to write H_k in (11) as

$$H_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}} \left[\begin{array}{c|c} 0 & \mathbb{0}^T \\ \hline \mathbb{0} & b_{k-1}^2 \mathbb{1}\mathbb{1}^T + I_{n-1} \end{array}\right] + \frac{(s_{k-1}^{(1)})^2}{s_{k-1}^T y_{k-1}} \left[\begin{array}{c|c} 1 & b_{k-1}\mathbb{1}^T \\ \hline b_{k-1}\mathbb{1} & b_{k-1}^2 \mathbb{1}\mathbb{1}^T \end{array}\right].$$

From (20) we can see that the fractions in front of the first and second matrices are the same, i.e.,

$$\frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}} = \frac{(s_{k-1}^{(1)})^2}{s_{k-1}^T y_{k-1}} = \frac{|s_{k-1}^{(1)}|}{2a}. \tag{22}$$

Hence, we obtain the following much more compact form

$$H_k = \gamma_k \left[\begin{array}{c|c} 1 & b_{k-1}\mathbb{1}^T \\ \hline b_{k-1}\mathbb{1} & 2b_{k-1}^2 \mathbb{1}\mathbb{1}^T + I_{n-1} \end{array}\right], \tag{23}$$

where

$$\gamma_k = \frac{|s_{k-1}^{(1)}|}{2a} \tag{24}$$

is the scale factor in (1). Finally, with the gradient defined in (15) we can compute the direction generated by Algorithm 1 in (12) as

$$d_k = -\frac{|s_{k-1}^{(1)}|}{2a} \begin{bmatrix} (-1)^k a + (n-1)b_{k-1} \\ \left((-1)^k a b_{k-1} + 2(n-1)b_{k-1}^2 + 1\right)\mathbb{1} \end{bmatrix}. \tag{25}$$

So, from definition (19) we can write b_k recursively as

$$b_k = \frac{(-1)^k a b_{k-1} + 2(n-1)b_{k-1}^2 + 1}{(-1)^k a + (n-1)b_{k-1}}. \tag{26}$$

2.1 *Existence of Armijo–Wolfe steps when* $\sqrt{3(n-1)} \le a$

In the next lemma we prove that if $\sqrt{3(n-1)} \le a$, then the $\{b_k\}$ alternate in sign with $|b_k| \le 1/a$.

LEMMA 2.1 Suppose $\sqrt{3(n-1)} \le a$. Define b_k as in (26) with $b_0 = 1/a$. Then $|b_k| \le 1/a$ and furthermore $\{b_k\}$ alternates in sign with

$$|b_k| = \frac{1 + (n-1)b_{k-1}^2}{a - (n-1)|b_{k-1}|} - |b_{k-1}|. \tag{27}$$

Proof. See Appendix A for the proof. □

Putting (26) and (27) together we can rewrite (25) as

$$d_k = -\frac{|s_{k-1}^{(1)}|}{2a}(a - (n-1)|b_{k-1}|)\begin{bmatrix} (-1)^k \\ |b_k|\mathbb{1} \end{bmatrix}. \tag{28}$$

Before stating the main result of this section we give the following simple lemma.

Lemma 2.2 Let $x \in \mathbb{R}^n$ be given, define

$$d_+ = -\begin{bmatrix} 1 \\ \beta\mathbb{1} \end{bmatrix} \text{and } d_- = -\begin{bmatrix} -1 \\ \beta\mathbb{1} \end{bmatrix}, \tag{29}$$

where $\beta > 0$, and define f by (2). Let d be either d_+ or d_-. Then $h(t) = f(x + td) - f(x)$ is unbounded below if and only if $\frac{a}{n-1} < \beta$.

Proof. We have

$$h(t) = a|x^{(1)} \pm t| - a|x^{(1)}| - (n-1)\beta t.$$

So,

$$(a - (n-1)\beta)\, t - 2a|x^{(1)}| < h(t) < (a - (n-1)\beta)\, t.$$

The result follows. □

Note that stating that h is unbounded below is not equivalent to saying that Armijo–Wolfe points do not exist along the direction d emanating from x. Such points may exist if the sign of $d^{(1)}$ is opposite to the sign of $x^{(1)}$.

Theorem 2.3 When Algorithm 1 is applied to (2) with $\sqrt{3(n-1)} \le a$, using any Armijo–Wolfe line search, with any starting point x_0 such that $x_0^{(1)} \neq 0$, the method generates directions d_k that are nonnegative scalar multiples of d_+ or d_-, defined in (29), with $\beta < a/(n-1)$. It follows that the steplength t_k satisfying the Armijo and Wolfe conditions (4) and (5) always exist and hence the method never terminates.

Proof. The proof is by induction on k. Without loss of generality assume $x_0^{(1)} > 0$, so $d_0 = -\nabla f(x_0) = ad_+$ with $\beta = 1/a$. Since $\sqrt{3(n-1)} \le a$, we have $1/a < a/(n-1)$ and hence the initial Armijo–Wolfe steplength t_0 exists by Lemma 2.2. Now, suppose that the result holds for all $j < k$, so d_k in (28) is well defined. Since by Lemma 2.1 we know that $|b_{k-1}| \le 1/a \le a/(n-1)$, the leading scalar in (28) is negative and therefore d_k is a nonnegative scalar multiple of d_+ or d_- with $\beta = |b_k| \le 1/a < a/(n-1)$. Hence, f is bounded below along the direction d_k emanating from x_k and so there exists t_k satisfying the Armijo and Wolfe conditions at iteration k, which implies that the algorithm does not terminate at iteration k. □

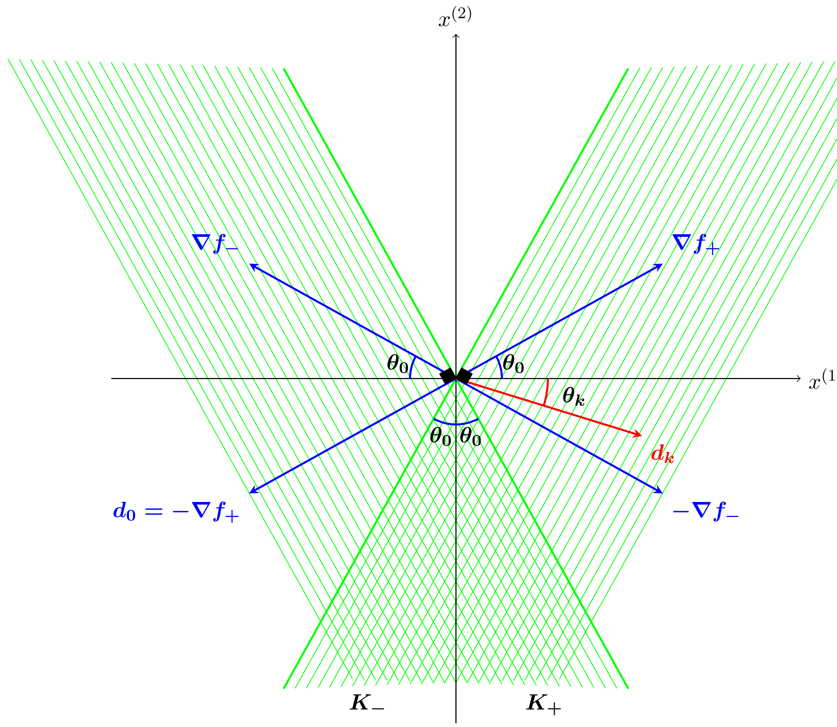

FIG. 1. **Angles of search directions.** Let $n = 2$, let $\nabla f_+ = [a1]^T$ and let $\nabla f_- = [-a1]^T$, so, since $x_0^{(1)} > 0$ by assumption, we have $d_0 = -\nabla f_+$. It follows from Lemma 2.1 that $b_k = d_k^{(2)}/d_k^{(1)}$ alternates in sign for $k = 1, 2, \ldots$, with absolute value bounded above by $1/a$, and hence that $\theta_k = \arctan(b_k)$ alternates in sign for $k = 1, 2, \ldots$, with $|\theta_k|$, the acute angle between the $x^{(1)}$ axis and the search direction d_k, bounded above by θ_0. Furthermore, Lemma 2.2 states that the function f is unbounded below along all directions in the open cones K_- and K_+, and bounded below along all other directions (except the vertical axis). Note, however, that points satisfying the Wolfe condition may exist along directions $d \in K_+$ emanating from iterates on the left side of the $x^{(2)}$ axis, but not along directions $d \in K_-$ emanating from the left side, because the former cross the $x^{(2)}$ axis and the latter do not, and vice versa. Finally, Theorem 2.3 implies that, under the assumption $a \geq \sqrt{3}$, we have $|\theta_k| \leq \theta_0 \leq \pi/6$, for all $k > 0$ (see the discussion after the theorem), so d_k does not lie in K_- or in K_+ and hence the algorithm does not terminate.

Using Fig. 1 we can provide an alternative informal geometrical proof for Theorem 2.3. We have

$$\frac{1}{a} \leq \frac{1}{\sqrt{3}} \Rightarrow \theta_0 = \arctan\frac{1}{a} \leq \arctan\frac{1}{\sqrt{3}} = \frac{\pi}{6}.$$

According to Lemma 2.1, we have $|b_k| \leq 1/a$, and so, $|\theta_k| \leq \theta_0$ and hence,

$$2\theta_0 + |\theta_k| \leq \frac{\pi}{2}.$$

It follows (see Fig. 1) that $d_k \notin K_+ \cup K_-$. This means that the method never generates a direction along which f is unbounded below.

However, Theorem 2.3 does not imply that Algorithm 1 converges to a nonoptimal point under the assumption that $\sqrt{3(n-1)} \leq a$, because the existence of Armijo–Wolfe steps t_k for all k does not imply that the sequence $\{f(x_k)\}$ is bounded below. This issue is addressed in the next section.

3. Failure of scaled memoryless BFGS

3.1 *Convergence of the absolute value of the normalized search direction when* $2\sqrt{n-1} \leq a$

Define

$$b = \frac{a - \sqrt{a^2 - 3(n-1)}}{3(n-1)} \tag{30}$$

and note that when $\sqrt{3(n-1)} \leq a$, then

$$\frac{1}{2a} \leq b \leq \frac{1}{a}.$$

Next we show the sequence $\{|b_k|\}$ converges to b under a slightly stronger assumption.

THEOREM 3.1 For $2\sqrt{n-1} \leq a$ the sequence defined by (27) converges and moreover

$$\lim_{k\to\infty} |b_k| = b.$$

Proof. See Appendix B for the proof. □

Note that the convergence result established in this theorem does not require any assumption of symmetry with respect to variables $2, 3, \ldots, n$ in the initial point x_0. The only assumption on x_0 is that $x_0^{(1)} > 0$. We need $x_0^{(1)} \neq 0$ so that f is differentiable at x_0; the assumption on the sign is purely for convenience.

ASSUMPTION 3.2 For the subsequent theoretical analysis we assume that

$$2\sqrt{n-1} \leq a.$$

With this assumption, as a direct implication of Theorem 3.1, for any given positive ϵ there exists K such that for $k \geq K$ we have

$$||b_k| - b| < \frac{\epsilon}{n-1}. \tag{31}$$

As we showed in Lemma 2.1, for $k \geq 0$ we have $|b_k| \leq 1/a$ and therefore

$$\frac{3(n-1)}{a} \leq a - \frac{n-1}{a} \leq a - (n-1)|b_k|. \tag{32}$$

Thus, $a - (n-1)|b_k|$ is positive and bounded away from zero.

Since $|b_k|$ converges by Theorem 3.1, we see that in the limit the normalized direction $d_k/\|d_k\|_2$ alternates between two limiting directions. It is this property that allows us to establish, under some subsequent assumptions, that scaled memoryless BFGS generates iterates x_k for which $f(x_k)$ is bounded below even though f is unbounded below.

3.2 *Dependence on the Armijo condition*

Combining (15) and (28) we get

$$\nabla f(x_k)^T d_k = -|d_k^{(1)}| \begin{bmatrix} (-1)^k a \\ \mathbb{1} \end{bmatrix}^T \begin{bmatrix} (-1)^k \\ |b_k|\mathbb{1} \end{bmatrix} = -|d_k^{(1)}| \left(a + (n-1)\,|b_k|\right), \tag{33}$$

so the Armijo condition (4) with $t = t_k$ at iteration k is

$$c_1 t_k |d_k^{(1)}| \left(a + (n-1)\,|b_k|\right) \leq f(x_k) - f(x_k + t_k d_k). \tag{34}$$

If t_k satisfies the Wolfe condition, i.e., t_k is large enough that the sign change (13) occurs, then we must have

$$|x_k^{(1)}| < t_k |d_k^{(1)}|. \tag{35}$$

Given this we can derive $f(x_k) - f(x_k + t_k d_k)$ using the definition of b_k in (19) as follows:

$$f(x_k) - f(x_k + t_k d_k) = 2a|x_k^{(1)}| - \left(a - (n-1)|b_k|\right) t_k |d_k^{(1)}|. \tag{36}$$

By defining φ_k as follows

$$\varphi_k = \frac{c_1 \left(a + (n-1)|b_k|\right) + a - (n-1)|b_k|}{2a}, \tag{37}$$

we can restate the Armijo condition in the following lemma.

LEMMA 3.3 Suppose t_k satisfies the Wolfe condition (13). Then for t_k to satisfy the Armijo condition (34) we must have

$$\varphi_k t_k |d_k^{(1)}| \leq |x_k^{(1)}|. \tag{38}$$

Proof. Combining (36) and (34) we get

$$c_1 t_k |d_k^{(1)}| \left(a + (n-1)\,|b_k|\right) \leq 2a|x_k^{(1)}| - \left(a - (n-1)\,|b_k|\right) t_k |d_k^{(1)}|,$$

and using the definition of φ_k in (37), (38) follows. □

From (35) and (38) we see that φ_k is the ratio of the lower bound and the upper bound on the steplength t_k provided by the Wolfe and Armijo conditions, respectively. The next lemma provides bounds on φ_k.

LEMMA 3.4

$$\frac{(n-1)|b_k|}{a} < \varphi_k. \tag{39}$$

Proof. Using Lemma 2.1 we know $3(n-1)|b_k| \leq a$ for all k, and so

$$2(n-1)|b_k| \leq a - (n-1)|b_k|,$$

and since

$$\frac{a-(n-1)|b_k|}{2a} = \varphi_k - c_1 \frac{a+(n-1)|b_k|}{2a},$$

and $c_1 > 0$, (39) follows. □

COROLLARY 3.5 For $k \geq 1$ we have

$$|s_k^{(1)}| \leq |s_{k-1}^{(1)}| \frac{1-\varphi_{k-1}}{\varphi_k}. \tag{40}$$

Proof. Summing the Armijo inequality (38) for two consecutive iterations we obtain

$$|s_{k-1}^{(1)}|\varphi_{k-1} + |s_k^{(1)}|\varphi_k \leq |x_{k-1}^{(1)}| + |x_k^{(1)}|,$$

and noticing that the right-hand side (R.H.S.), according to (14), is equal to $|s_{k-1}^{(1)}|$ we get (40). □

LEMMA 3.6 For any given $\epsilon > 0$ let K be the smallest integer such that for any $k \geq K$, (31) holds. Then for all $N > K$ we have

$$f(x_K) - f(x_N) < a|x_K^{(1)}| + ((n-1)\,b + \epsilon) \sum_{k=K}^{N-1} |s_k^{(1)}|. \tag{41}$$

Proof. Using $t_k d_k = s_k$ and $x_{k+1} = x_k + s_k$ in (36) and then applying (31) we obtain

$$f(x_k) - f(x_{k+1}) < 2a|x_k^{(1)}| - a|s_k^{(1)}| + ((n-1)\,b + \epsilon)\,|s_k^{(1)}|. \tag{42}$$

Summing up (42) from $k = K$ to $k = N-1$ and recalling (14), we get

$$f(x_K) - f(x_N) < a\sum_{k=K}^{N-1} |s_k^{(1)}| + a|x_K^{(1)}| - a|x_N^{(1)}| - a\sum_{k=K}^{N-1} |s_k^{(1)}| + ((n-1)\,b + \epsilon)\sum_{k=K}^{N-1} |s_k^{(1)}|.$$

Canceling the first and fourth terms and dropping $-a|x_N|$, we arrive at (41). □

From applying Theorem 3.1 to the definition of φ_k in (37) it is immediate that $\{\varphi_k\}$ converges. Let

$$\varphi = \frac{c_1\,(a+(n-1)b) + a - (n-1)b}{2a}, \tag{43}$$

so

$$\lim_{k\to\infty} \varphi_k = \varphi. \tag{44}$$

LEMMA 3.7 Assume

$$0 < \epsilon \le \frac{\sqrt{a^2 - 3(n-1)}}{3}, \tag{45}$$

and let K be defined as in Lemma 3.6. Then for any $k \ge K$ we have

$$\left| \frac{1-\varphi_{k-1}}{\varphi_k} - \frac{1-\varphi}{\varphi} \right| < \frac{15}{a}\epsilon. \tag{46}$$

Proof. By rearranging terms in (30) and using (45) we get

$$(n-1)b + \epsilon \le (n-1)b + \frac{\sqrt{a^2 - 3(n-1)}}{3} = \frac{a}{3}. \tag{47}$$

Using (31) and (47), for $k \ge K$ we have

$$0 < a - (n-1)b - \epsilon < a - (n-1)|b_k|.$$

Combining this with (39) we get

$$0 < \frac{a-(n-1)b-\epsilon}{2a} < \varphi_k < 1.$$

Hence,

$$1 < \frac{1}{\varphi_k} < \frac{2a}{a-(n-1)b-\epsilon} \le \frac{2a}{a-\frac{a}{3}} = 3.$$

Since $0 < c_1 < 1$, from (31), (37), (43) and (44) we get

$$|\varphi_k - \varphi| < \frac{(1+c_1)\epsilon}{2a} < \frac{\epsilon}{a}.$$

So,

$$\left| \frac{1-\varphi_{k-1}}{\varphi_k} - \frac{1-\varphi}{\varphi} \right| = \left| \frac{1}{\varphi_k} - 1 + \frac{\varphi_k - \varphi_{k-1}}{\varphi_k} - \frac{1}{\varphi} + 1 \right|$$
$$< \left| \frac{\varphi - \varphi_k}{\varphi_k \varphi} \right| + \left| \frac{\varphi_k - \varphi_{k-1}}{\varphi_k} \right| < \frac{\epsilon}{a\varphi_k}\left(\frac{1}{\varphi} + 2\right).$$

Note that $1 < 1/\varphi_k < 3$ applies to all φ_k (as well as the limit φ) with $k \ge K$, and therefore we conclude (46). □

Let

$$\psi_\epsilon = \frac{1-\varphi}{\varphi} + \frac{15}{a}\epsilon. \tag{48}$$

If Lemma 3.7 applies then from (40) and (46) we conclude

$$|s_k^{(1)}| < \psi_\epsilon |s_{k-1}^{(1)}|. \tag{49}$$

That is to say, with ϵ satisfying (45), after at most K iterations, (49) holds. Consequently, with the additional assumption $\psi_\epsilon < 1$, we obtain

$$\sum_{k=K}^{N-1} |s_k^{(1)}| < |s_K^{(1)}| \frac{1}{1-\psi_\epsilon}. \tag{50}$$

Now we can prove the main result of this subsection. Recall that $c_1 < 1$.

THEOREM 3.8 Suppose c_1 is chosen large enough that

$$\frac{1}{c_1} - 1 < \frac{a}{(n-1)b} \tag{51}$$

holds. Then, using any Armijo–Wolfe line search with any starting point x_0 with $x_0^{(1)} \neq 0$, scaled memoryless BFGS applied to (2) fails in the sense that $f(x_N)$ is bounded below as $N \to \infty$.

Proof. It follows from (51) and (43) that $\varphi > 1/2$. Therefore, using (48), we can choose ϵ small enough such that $\psi_\epsilon < 1$ holds in addition to (45). Applying Lemmas 3.6 and 3.7, we conclude that there exists K such that for any $N > K$, (50) holds, and, substituting this into (41) we get

$$f(x_K) - f(x_N) < a|x_K^{(1)}| + |s_K^{(1)}| \frac{(n-1)\,b+\epsilon}{1-\psi_\epsilon}. \tag{52}$$

This establishes that $f(x_N)$ is bounded below for all $N > K$. □

Using (30) we see that the failure condition (51) for scaled memoryless BFGS with any Armijo–Wolfe line search applied to (2) is equivalent to

$$\frac{1-c_1}{c_1}(n-1) < a^2 + a\sqrt{a^2 - 3(n-1)}. \tag{53}$$

The corresponding failure condition for the gradient method on the same function, again using any Armijo–Wolfe line search, is, as we showed in Asl & Overton (2019),

$$\frac{1-c_1}{c_1}(n-1) < a^2. \tag{54}$$

Hence, scaled memoryless BFGS fails under a *weaker* condition relating a to the Armijo parameter than the condition for failure of the gradient method on the same function with the same line search conditions. Indeed, Assumption 3.2 implies

$$a^2 + a\sqrt{a^2 - 3(n-1)} \geq 4(n-1) + 2\sqrt{n-1}\sqrt{n-1} = 6(n-1).$$

So, if the Armijo parameter $c_1 \geq 1/7$, then (53) holds. In contrast, the same assumption implies that if $c_1 \geq 1/5$, then (54) holds. So, scaled memoryless BFGS with any Armijo–Wolfe line search applied to (2) fails under a weaker condition on the Armijo parameter than the gradient method does.

3.3 *Results for a specific Armijo–Wolfe line search, independent of the Armijo parameter*

Considering only the first component of the direction d_k in (28) we have

$$\frac{2a}{a-(n-1)|b_{k-1}|}|d_k^{(1)}| = |s_{k-1}^{(1)}|. \tag{55}$$

Using (17), it follows that if

$$t_k < \frac{2a}{a-(n-1)|b_{k-1}|}, \tag{56}$$

we have $|s_k^{(1)}| < |s_{k-1}^{(1)}|$. Note that the R.H.S. of (56) is greater than two. However, as shown in the next lemma, except at the initial iteration ($k=0$), $t=2$ is always large enough to satisfy the Wolfe condition, implying that there exists $t \leq 2$ satisfying both the Armijo and Wolfe conditions.

Lemma 3.9 For $k \geq 1$, the steplength $t_k = 2$ always satisfies the Wolfe condition (13), i.e., we have

$$|x_k^{(1)}| < 2|d_k^{(1)}|. \tag{57}$$

Proof. Since $k \geq 1$, we know that the Armijo and Wolfe conditions hold at iteration $k-1$ by definition of Algorithm 1. So, using (38) and (17) we have

$$\varphi_{k-1}|s_{k-1}^{(1)}| \leq |x_{k-1}^{(1)}|. \tag{58}$$

Using the inequality (39) in the left-hand side (L.H.S.) and the equality (14) in the R.H.S. we get

$$\frac{(n-1)|b_{k-1}|}{a}|s_{k-1}^{(1)}| < |s_{k-1}^{(1)}| - |x_k^{(1)}|,$$

i.e.,

$$|x_k^{(1)}| < |s_{k-1}^{(1)}|\frac{a-(n-1)|b_{k-1}|}{a}.$$

Substituting (55) into the R.H.S., we obtain (57). □

Now let us focus on the Armijo–Wolfe bracketing line search given in Lewis & Overton (2013) and Asl & Overton (2019), which we state here for convenience.

Algorithm 2 (Armijo–Wolfe bracketing line search)

$\alpha \leftarrow 0$
$\beta \leftarrow +\infty$
$t \leftarrow 1$
while true **do**
  if the Armijo condition (4) fails **then**
    $\beta \leftarrow t$
  else if the Wolfe condition (5) fails **then**
    $\alpha \leftarrow t$
  else
    stop and return t
  end if
  if $\beta < +\infty$ **then**
    $t \leftarrow (\alpha + \beta)/2$
  else
    $t \leftarrow 2\alpha$
  end if
end while

It is known from the results in Lewis & Overton (2013) that provided f is bounded below along d_{k-1} (as we already established must hold for directions generated by Algorithm 1), the Armijo–Wolfe bracketing line search will terminate with a steplength t satisfying both conditions. In the following lemma we show that if we use this line search, it always generates $t_k \leq 2$ for $k \geq 1$.

LEMMA 3.10 When scaled memoryless BFGS is applied to (2), using Algorithm 2 it always returns steplength $t_k \leq 2$ for $k \geq 1$.

Proof. The line search begins with the unit step. If this step, $t = 1$, does not satisfy the Armijo condition (4), then the step is contracted, so the final step is less than one. On the other hand, if $t = 1$ satisfies (4), then the line search checks whether the Wolfe condition (5) is satisfied too. If it is, then the line search quits; if not, the step is doubled and hence the line search next checks whether $t = 2$ satisfies (5). At the initial iteration ($k = 0$), several doublings might be needed before (5) is eventually satisfied. But for subsequent steps ($k \geq 1$), we know that $t = 2$ must satisfy the Wolfe condition, so the final step must satisfy $t_k = 2$ (if $t = 2$ satisfies (4)) or $t_k < 2$ (otherwise). Thus, for $k \geq 1$ we always have $t_k \leq 2$. □

Now we can present the main result of this subsection; using a line search with the property just described, the optimization method fails.

THEOREM 3.11 If scaled memoryless BFGS is applied to (2), using an Armijo–Wolfe line search that satisfies $t_k \leq 2$ for $k \geq 1$, such as Algorithm 2, then the method fails in the sense that $f(x_N)$ is bounded below as $N \to \infty$.

Proof. Recalling $t_{k+1} d_{k+1}^{(1)} = s_{k+1}^{(1)}$ again, using (55) and $t_{k+1} \leq 2$ we find that

$$|s_{k+1}^{(1)}| \leq \frac{a - (n-1)|b_k|}{a} |s_k^{(1)}|. \tag{59}$$

Let $\epsilon > 0$ satisfy

$$\delta_\epsilon \equiv \frac{a-(n-1)b}{a} + \frac{\epsilon}{a} < 1.$$

Define K as in Lemma 3.6, so that (31) holds, and hence

$$\frac{a-(n-1)|b_k|}{a} < \delta_\epsilon.$$

Applying this inequality to (59) we get

$$|s_{k+1}^{(1)}| \le \delta_\epsilon |s_k^{(1)}|, \tag{60}$$

and since $\delta_\epsilon < 1$ we have

$$\sum_{k=K}^{N-1} |s_k^{(1)}| < |s_K^{(1)}| \frac{1}{1-\delta_\epsilon}. \tag{61}$$

By substituting this into (41) we get

$$f(x_K) - f(x_N) < a|x_K^{(1)}| + |s_K^{(1)}| \frac{(n-1)\,b+\epsilon}{1-\delta_\epsilon},$$

which shows $f(x_N)$ is bounded below. □

Finally, we have the following corollary to Theorems 3.8 and 3.11. Recall that γ_k is the scale factor (see (24)).

COROLLARY 3.12 If the assumptions required by either Theorem 3.8 or 3.11 hold, then

$$\lim_{N\to\infty} \gamma_N = 0 \tag{62}$$

and x_N converges to a nonoptimal point $\bar{x}$ such that

$$\bar{x} = [0, \bar{x}^{(2)}, \ldots, \bar{x}^{(n)}]^T. \tag{63}$$

Proof. It is immediate from (50) or (61) that $|s_N^{(1)}| \to 0$ as $N \to \infty$, so from (24), we conclude (62). Also due to (14) we have $|x_N^{(1)}| \to 0$, and since $f(x_N) = a|x_N^{(1)}| + \sum_{i=2}^{n-1} x_N^{(i)}$ is bounded below, so is $\sum_{i=2}^{n-1} x_N^{(i)}$. Due to (32) and (28), we have $d_{N-1}^{(i)} < 0$, for $i = 2, 3, \ldots, n$, so $t_{N-1} d_{N-1}^{(i)} = x_N^{(i)} - x_{N-1}^{(i)} < 0$, and therefore $x_N^{(i)}$ is strictly decreasing as $N \to \infty$. Hence, $x_N^{(i)}$ converges to a limit $\bar{x}^{(i)}$. □

Due to the symmetry we discussed earlier, the total decrease along each component, $x_0^{(i)} - \bar{x}^{(i)} = \sum_{k=0}^{N} s_k^{(i)}$, is the same for $i = 2, 3, \ldots, n$.

Finally, note that it follows from Corollary 3.12 together with (23) that, when the assumptions hold, the matrix H_N converges to zero. In contrast, when full BFGS is applied to the same problem, it is typically the case that a direction is identified along which f is unbounded below within a few iterations, and that at the final iterate, one eigenvalue of the inverse Hessian is much smaller than the others, with its corresponding eigenvector close to the first coordinate vector, along which f is nonsmooth.

4. Experiments

Our experiments were conducted using the BFGS / L-BFGS MATLAB code in HANSO.[1] This uses the Armijo–Wolfe bracketing line search given in Algorithm 2. Consequently, according to the results of Section 3.3, scaled memoryless BFGS (L-BFGS with $m = 1$) should fail on function (2) when a satisfies Assumption 3.2: $2\sqrt{n-1} \leq a$. This is illustrated in Fig. 2, which shows an experiment where we set $a = 3$ and $n = 2$ and ran scaled memoryless BFGS, the gradient method and full BFGS, starting from the same randomly generated initial point. We see that scaled memoryless BFGS fails, in the sense that it converges to a nonoptimal point, while the gradient method succeeds, in the sense that it generates iterates with $f(x_k) \downarrow -\infty$. In contrast to both, full BFGS succeeds in the sense that it finds a direction along which f is unbounded below in just five iterations. These three different outcomes respectively illustrate the three different ways that the HANSO code terminated in our experiments: (i) convergence to a nonoptimal point, which is detected when the steplength upper bound β in Algorithm 2 converges to zero indicating that Armijo–Wolfe points exist, but the line search terminates without finding one due to rounding errors; (ii) divergence of the $f(x_k)$ to $-\infty$ although the line search always finds Armijo–Wolfe steplengths; and (iii) generation of a direction along which f is apparently unbounded below, which is detected when β in Algorithm 2 remains equal to its initial value of ∞ while the lower bound α is repeatedly doubled until a limit is exceeded.[2] In the results reported below for function (2), termination (i) is considered a failure while terminations (ii) and (iii) are considered successes. We note that, provided $\sqrt{n-1} \leq a$, the gradient method can never result in termination (iii), and whether it results in termination (i) or (ii) depends on the Armijo parameter (Asl & Overton, 2019). In our experiments, L-BFGS, with or without scaling and with one or more updates, always resulted in termination (i) or (iii), while full BFGS invariably resulted in termination (iii) (as we know it must from the results in Xie & Waechter, 2017).

Although the proof of Theorem 3.1 does require Assumption 3.2 we observed that $\sqrt{3(n-1)} \leq a$ suffices for $\{|b_k|\}$ and consequently $|d_k|/\|d_k\|_2$ to converge. In Fig. 3 we repeat the same experiment with $a = \sqrt{3}$ and $n = 2$, showing that scaled memoryless BFGS still fails. In this case, as noted in Section 3, the normalized direction is the same as the normalized direction generated by the gradient method, but unlike in the gradient method, the magnitude of the directions d_k converge to zero so scaled memoryless BFGS fails.

However, if we set a to $\sqrt{3} - 0.001$ the method succeeds. This is demonstrated in Fig. 4; observe that although one at first has the impression that x_k is converging to a nonoptimal point, a search direction is generated on which f is unbounded below 'at the last minute'.

Extensive additional experiments verify that the condition $\sqrt{3(n-1)} \leq a$, as opposed to Assumption 3.2, is sufficient for failure, as illustrated by the magenta asterisks in Fig. 5. Starting from 5000 random points generated from the normal distribution, we called scaled memoryless BFGS to minimize function (2) with $n = 30$ and for values of a ranging from 9.317 to 9.337, since for $n = 30$, $\sqrt{3(n-1)} \approx 9.327$. We see that for $9.327 \leq a$ the failure rate is 1 (100%), while for $9.32 > a$ the failure rate is 0. In comparison to a similar experiment in Asl & Overton (2019) for the gradient method,

[1] www.cs.nyu.edu/overton/software/hanso/

[2] Although in principle the code would alternatively terminate if a termination tolerance was met or an upper bound on the number of iterations was exceeded, we set these so small and large, respectively, that they virtually never caused termination.

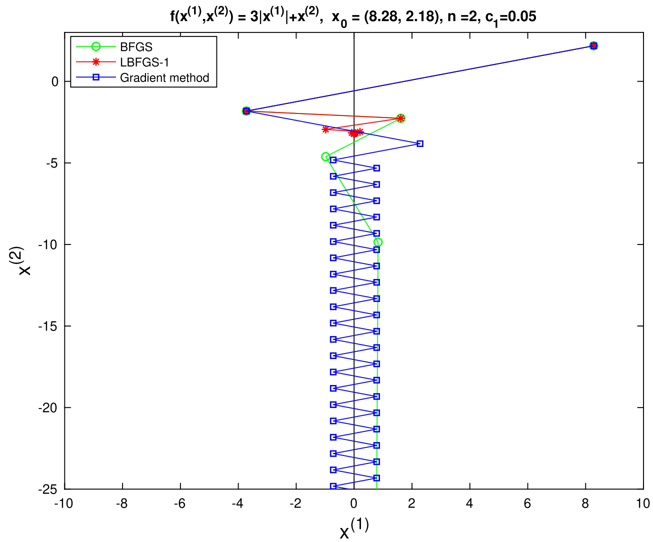

FIG. 2. Full BFGS (green circles), scaled memoryless BFGS (red asterisks) and the gradient method (blue squares) applied to the function (2) defined by $a = 3$ and $n = 2$. Scaled memoryless BFGS fails while full BFGS and the gradient method succeed.

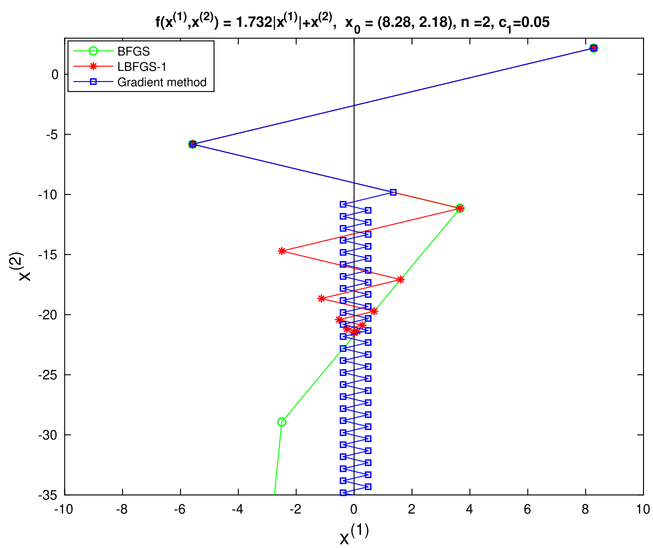

FIG. 3. Full BFGS (green circles), scaled memoryless BFGS (red asterisks) and the gradient method (blue squares) applied to the function (2) defined by $a = \sqrt{3}$ and $n = 2$. Scaled memoryless BFGS fails while full BFGS and the gradient method succeed.

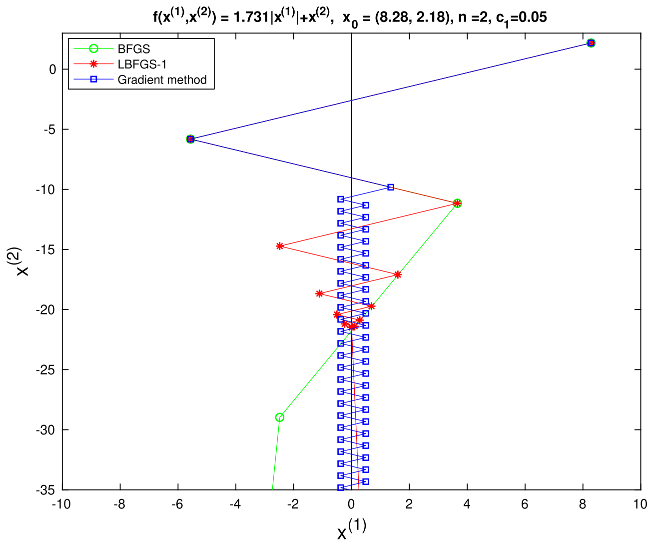

FIG. 4. Full BFGS (green circles), scaled memoryless BFGS (red asterisks) and the gradient method (blue squares) applied to the function (2) defined by $a = \sqrt{3} - 0.001$ and $n = 2$. All methods succeed.

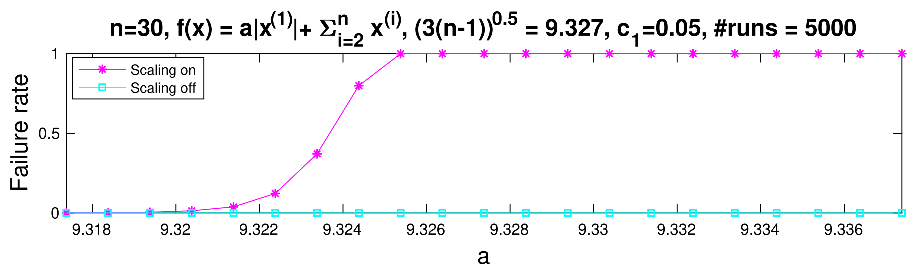

FIG. 5. The failure rate of memoryless BFGS with scaling (magenta asterisks) and without scaling (cyan squares) applied to function (2) with $n = 30$ and 21 different values of a, initiating the method from 5000 random points. With scaling, the failure rate is 1 for $9.327 \leq a$. Without scaling, the failure rate is 0 regardless of a.

the transition from failure rate 0 to 1 is quite sharp here. This might be explained by the fact that the gradient method fails because the steplength $t_k \to 0$, whereas for scaled memoryless BFGS, t_k does not converge to zero; it is the scale γ_k and consequently the norm of d_k which converges to zero. Hence, rounding error prevents the observation of a sharp transition in the results for the gradient method, as explained in Asl & Overton (2019); by comparison, rounding error plays a less significant role in the experiments reported here.

The cyan squares in Fig. 5 show the results from the same experiment for memoryless BFGS *without* scaling, i.e., with $H_k^0 = I$ instead of (1), using the same 5000 initial points. In this case, the method is successful regardless of the value of a.

Experiments suggest that the theoretical results we presented for scaled L-BFGS with only one update might extend, although undoubtedly in a far more complicated form, to any number of updates. In Fig. 6 we show results of experiments with a variety of choices of m and a, running scaled L-BFGS-m (L-BFGS with m updates) initiated from 1000 randomly generated points for each pair (m,a). The

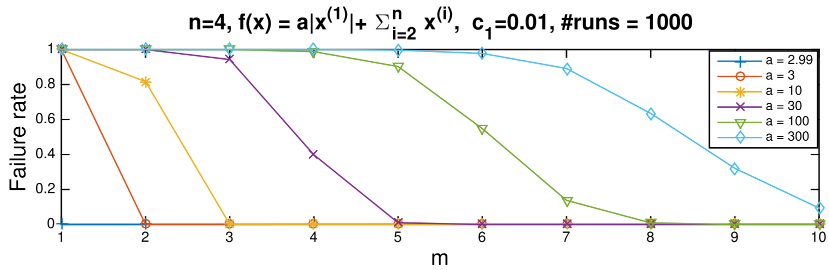

FIG. 6. The failure rate for each scaled L-BFGS-m, where the number of updates m ranges from 1 to 10, applied to function (2) with $a = 2.99$ (blue pluses), $a = 3$ (orange circles), $a = 10$ (yellow asterisks), $a = 30$ (purple crosses), $a = 100$ (green triangles) and finally $a = 300$ (cyan diamonds), with $c_1 = 0.01$ and $n = 4$ and hence $\sqrt{3(n-1)} = 3$, and with each experiment initiated from 1000 random points.

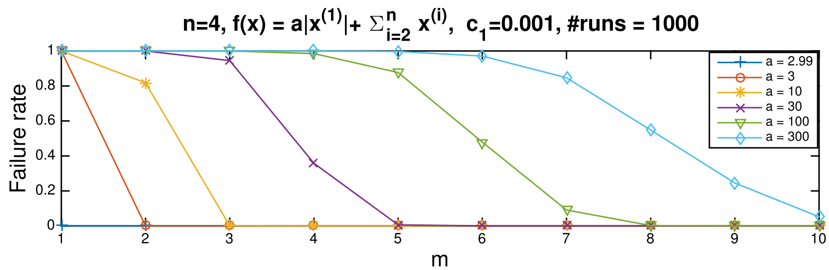

FIG. 7. The same experiment as in Fig. 6 except that $c_1 = 0.001$.

horizontal axis shows m, the number of updates, while the vertical axis shows the observed failure rate. We set the Armijo parameter $c_1 = 0.01$ and $n = 4$, so that $\sqrt{3(n-1)} = 3$, and show results for values of a ranging from 2.99 to 300. Figure 7 shows results from the same experiment except that $c_1 = 0.001$. The results shown in Fig. 8 use a different objective function; instead of (2), we define $f(x) = a|b_1^T x| + b_2^T x$, where b_1 and b_2 were each chosen as a random vector in $\mathbb{R}^{10}$ and normalized to have length one. The Armijo parameter was set to $c_1 = 0.01$. In all of Figs 6, 7 and 8 we observe that as a gets larger for a fixed m, the failure rate increases. On the other hand, as m gets larger for a fixed a, the failure rate decreases. Comparing Figs 6 and 7, we see that the results do not demonstrate a significant dependence on the Armijo parameter c_1; in particular, as we established in Section 3.3, there is no dependence on c_1 when $m = 1$ because we are using the line search in Algorithm 2. However, we do observe small differences for the larger values of m, where the failure rate is slightly higher for the larger Armijo parameter. This is consistent with the theoretical results in Section 3.2 as well as those in Asl & Overton (2019), where, if a is relatively large, then to avoid failure c_1 should not be too large.

Finally, we conducted experiments with a more general class of piecewise linear max functions defined as

$$f(x) = \max_{i=1,\ldots p} \{b_i^T x - r_i\}, \tag{64}$$

where $b_1, \ldots, b_p$ are randomly generated vectors in $\mathbb{R}^n$ and $r_1, \ldots, r_p$ are random scalars. These quantities were fixed for the experiment reported here but similar results were obtained for other choices. We set $n = 10$ and $p = 50$, obtaining a problem that, unlike those studied above, is bounded below.

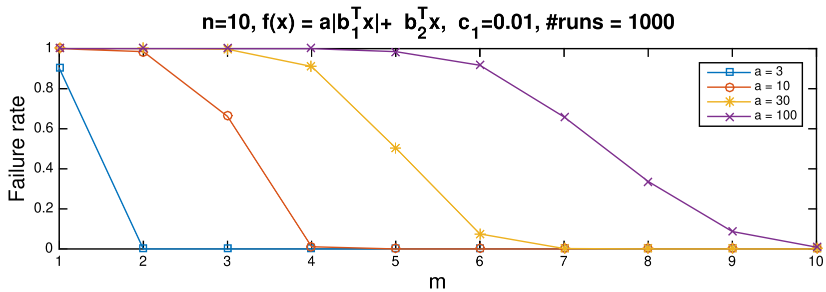

FIG. 8. The same experiment as in Fig. 6 except that $f(x) = a|b_1^T x| + b_2^T x$ where $b_1, b_2 \in \mathbb{R}^{10}$ were chosen randomly.

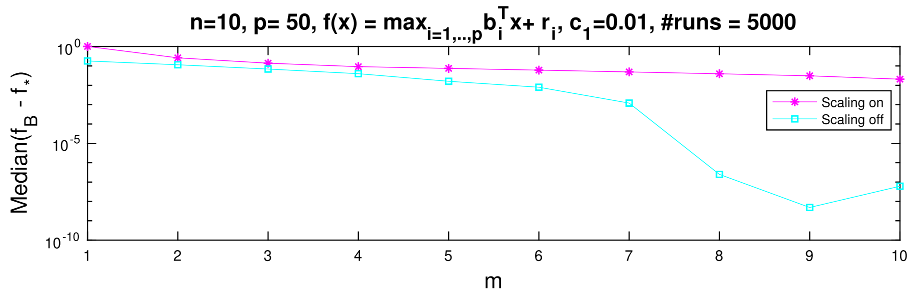

FIG. 9. Median accuracy of the solution f_B found by L-BFGS-m with $m = 1, \ldots, 10$ for the piecewise linear function defined in (64), with $n = 10$ and $p = 50$, compared with the value f_* obtained from the linear optimizer in MOSEK using high accuracy. Scaled L-BFGS-m does not obtain accurate solutions even with $m = 10$. In contrast, with scaling off, L-BFGS-9 obtains a median accuracy of about 10^{-9}.

Consequently, all runs result in termination (i), and we evaluated how successful they were by comparing the final function value to the optimal value f_* that we obtained via linear programming using MOSEK[3] with the tolerance set to 10^{-14}. Figure 9 shows the median accuracy obtained by L-BFGS-m, for $m = 1, \ldots, 10$, with and without scaling. L-BFGS with scaling does not achieve a median accuracy better than 10^{-2}, even when $m = 10$. Without scaling, the accuracy of the results improves substantially, to a median accuracy of about 10^{-9} with $m = 9$. Strangely, for this problem, and many different instances of it that we tried, L-BFGS-10 performs worse than L-BFGS-9. The median accuracy of the solution found by full BFGS (with or without scaling the initial inverse Hessian approximation) is significantly better: about 10^{-14}.

5. Concluding remarks

We have given the first analysis of a variant of L-BFGS applied to a nonsmooth function, showing that the scaled version of memoryless BFGS (L-BFGS with just one update) applied to (2) generates iterates converging to a nonoptimal point under simple conditions. One of these conditions applies to the method with any Armijo–Wolfe line search and depends on the Armijo parameter. The other condition applies to the method using a standard Armijo–Wolfe bracketing line search and does not depend on the

[3] https://www.mosek.com/

Armijo parameter. Experiments suggest that extended results likely hold for L-BFGS with more than one update, though clearly a generalized analysis would be much more complicated.

We do not know whether L-BFGS without scaling applied to the same function can converge to a nonoptimal point, but numerical experiments suggest that this cannot happen. Furthermore, we observed that L-BFGS without scaling obtains significantly more accurate solutions than L-BFGS with scaling when applied to a more general piecewise linear function that is bounded below. Nonetheless, it remains an open question as to whether scaling is generally inadvisable when applying L-BFGS to nonsmooth functions, despite its apparent advantage for smooth optimization.

Acknowledgements

Many thanks to Margaret H. Wright for arranging financial support for the first author from the Simons Foundation (417314, MHW). Thanks also to the anonymous referees for carefully reading the paper and suggesting several improvements.

Funding

National Science Foundation (DMS-1620083 to M.L.O.).

References

Asl, A. & Overton, M. L. (2019) Analysis of the gradient method with an Armijo–Wolfe line search on a class of nonsmooth convex functions. *Optim. Methods Softw.*, **35**, 223–242.

Clarke, F. H. (1990) *Optimization and Nonsmooth Analysis*, 2nd edn. Classics in Applied Mathematics, vol. 5. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

Curtis, F. E., Mitchell, T. & Overton, M. L. (2017) A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. *Optim. Methods Softw.*, **32**, 148–181.

Dai, Y.-H. (2002) Convergence properties of the BFGS algorithm. *SIAM J. Optim.*, **13**, 693–701.

Gill, P. E. & Leonard, M. W. (2003) Limited-memory reduced-Hessian methods for large-scale unconstrained optimization. *SIAM J. Optim.*, **14**, 380–401.

Guo, J. & Lewis, A. (2018) Nonsmooth variants of Powell's BFGS convergence theorem. *SIAM J. Optim.*, **28**, 1301–1311.

Le, Q. V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B. & Ng, A. Y. (2011) On optimization methods for deep learning. *Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11*. USA: Omnipress, pp. 265–272.

Lewis, A. S. & Overton, M. L. (2013) Nonsmooth optimization via quasi-Newton methods. *Math. Program.*, **141**, 135–163.

Lewis, A. S. & Zhang, S. (2015) Nonsmoothness and a variable metric method. *J. Optim. Theory Appl.*, **165**, 151–171.

Lin, H., Mairal, J. & Harchaoui, Z. (2016) An inexact variable metric proximal point algorithm for generic quasi-Newton acceleration. *SIAM J. Optim.*, **29**, 1408–1443.

Liu, D. C. & Nocedal, J. (1989) On the limited memory BFGS method for large scale optimization. *Math. Program.*, **45**, 503–528.

Mascarenhas, W. F. (2004) The BFGS method with exact line searches fails for non-convex objective functions. *Math. Program.*, **99**, 49–61.

Mokhtari, A. & Ribeiro, A. (2015) Global convergence of online limited memory BFGS. *J. Mach. Learn. Res.*, **16**, 3151–3181.

Nocedal, J. & Wright, S. J. (2006) *Numerical Optimization*, 2nd edn. New York: Springer.

Powell, M. J. D. (1976) Some global convergence properties of a variable metric algorithm for minimization without exact line searches. *Nonlinear Programming*. SIAM-AMS Proceedings, vol. 9. Providence: American Mathematical Society, pp. 53–72.

Xie, Y. & Waechter, A. (2017) On the convergence of BFGS on a class of piecewise linear non-smooth functions, *Preprint* arXiv:1712.08571.

A. Proof of Lemma 2.1

Suppose $\sqrt{3(n-1)} \leq a$. Using a change of variable such that $\beta_k = b_k$ when k is even, and $\beta_k = -b_k$ when k is odd, (26) becomes

$$\beta_k = \frac{1+(n-1)\beta_{k-1}^2}{a-(n-1)\beta_{k-1}} - \beta_{k-1}. \tag{A.1}$$

From (18) we have $\beta_0 = 1/a$. Using induction we prove that $0 < \beta_k \leq 1/a$. This is clearly true for $k = 0$. Suppose we have $0 < \beta_{k-1} \leq 1/a$. Hence,

$$\beta_{k-1} < \frac{1}{a-(n-1)\beta_{k-1}} < \frac{1+(n-1)\beta_{k-1}^2}{a-(n-1)\beta_{k-1}},$$

so, dropping the middle term and moving β_{k-1} to the R.H.S., we get exactly the definition of β_k according to (A.1). So, we have $0 < \beta_k$. Next, starting from $\sqrt{3(n-1)} \leq a$, we show that $\beta_k \leq 1/a$:

$$\begin{aligned}
&\frac{3(n-1)}{a} \leq a \Rightarrow \\
&\frac{(n-1)}{a} + 2(n-1)\beta_{k-1} \leq a \Rightarrow \\
&\frac{a^2+n-1}{a} \leq 2(a-(n-1)\beta_{k-1}) \Rightarrow \\
&\frac{a^2+n-1}{a(a-(n-1)\beta_{k-1})} \leq 2.
\end{aligned}$$

Multiplying both sides by β_{k-1} we get

$$\frac{a\beta_{k-1}+1}{a-(n-1)\beta_{k-1}} - \frac{1}{a} \leq 2\beta_{k-1},$$

and finally by moving $1/a$ to the right and $2\beta_{k-1}$ to the left we get

$$\frac{1+(n-1)\beta_{k-1}^2}{a-(n-1)\beta_{k-1}} - \beta_{k-1} \leq \frac{1}{a}.$$

The L.H.S. is β_k as it's defined in (A.1), so $\beta_k \leq 1/a$. Recalling the change of variable in the beginning of the proof it follows that $\beta_k = |b_k|$. So, from (A.1) we get (27).

B. Proof of Theorem 3.1

We continue to use the same change of variable as before, that is $\beta_k = b_k$ when k is even, and $\beta_k = -b_k$ when k is odd. In this way, (A.1) is equivalent to (27), and we prove that if $2\sqrt{n-1} \le a$, then $\{\beta_k\}$ converges. From a little rearrangement in (A.1) we can easily get

$$a(\beta_k + \beta_{k-1}) = 1 + 2(n-1)\beta_{k-1}^2 + (n-1)\beta_{k-1}\beta_k, \tag{B.1}$$

and by moving $(n-1)\beta_{k-1}\beta_k$ to the left and adding 1 to both sides we get

$$a(\beta_k + \beta_{k-1}) - (n-1)\beta_{k-1}\beta_k + 1 = 2\big(1 + (n-1)\beta_{k-1}^2\big). \tag{B.2}$$

For further simplification we define

$$\rho_k = \frac{1 + (n-1)\beta_k^2}{a - (n-1)\beta_k}, \tag{B.3}$$

so we can rewrite (A.1) as

$$\beta_{k+1} = \rho_k - \beta_k. \tag{B.4}$$

By applying (B.4) recursively we obtain

$$\beta_{k+1} - \beta_{k-1} = \rho_k - \rho_{k-1}. \tag{B.5}$$

Note that from (B.3) we have

$$\begin{aligned} \rho_k - \rho_{k-1} &= \frac{1 + (n-1)\beta_k^2}{a - (n-1)\beta_k} - \frac{1 + (n-1)\beta_{k-1}^2}{a - (n-1)\beta_{k-1}} \\ &= \frac{\Big(1 + (n-1)\beta_k^2\Big)\Big(a - (n-1)\beta_{k-1}\Big) - \Big(1 + (n-1)\beta_{k-1}^2\Big)\Big(a - (n-1)\beta_k\Big)}{\Big(a - (n-1)\beta_k\Big)\Big(a - (n-1)\beta_{k-1}\Big)} \\ &= \frac{(\beta_k - \beta_{k-1})(n-1)\Big(a(\beta_k + \beta_{k-1}) - (n-1)\beta_{k-1}\beta_k + 1\Big)}{\Big(a - (n-1)\beta_k\Big)\Big(a - (n-1)\beta_{k-1}\Big)}. \end{aligned} \tag{B.6}$$

The last factor in the numerator is the L.H.S. in (B.2), so

$$\rho_k - \rho_{k-1} = \frac{(\beta_k - \beta_{k-1})(n-1)2\Big(1 + (n-1)\beta_{k-1}^2\Big)}{\Big(a - (n-1)\beta_k\Big)\Big(a - (n-1)\beta_{k-1}\Big)}. \tag{B.7}$$

Hence, since all of the factors in this product except $(\beta_k - \beta_{k-1})$ are known to be positive, we have

$$(\rho_k - \rho_{k-1})(\beta_k - \beta_{k-1}) \ge 0. \tag{B.8}$$

Putting (B.5) and (B.8) together we conclude

$$(\beta_{k+1} - \beta_{k-1})(\beta_k - \beta_{k-1}) \geq 0. \tag{B.9}$$

As the next step we will show that

$$(\beta_{k+1} - \beta_k)(\beta_k - \beta_{k-1}) \leq 0. \tag{B.10}$$

Since $a \geq 2\sqrt{n-1}$ and using $1/a \geq \beta_{k-1}$ we get

$$\begin{aligned}
&\big(a^2 - 4(n-1)\big)\big(a^2 + (n-1)\big) \geq 0 \Rightarrow \\
&a^2 - 3(n-1) \geq \frac{4(n-1)^2}{a^2} \Rightarrow \\
&a^2 - 3(n-1) \geq 4(n-1)^2\beta_{k-1}^2 \Rightarrow \\
&a^2 - 3(n-1) - 4(n-1)^2\beta_{k-1}^2 \geq 0.
\end{aligned}$$

By adding and deducting $2(n-1)^2\beta_k\beta_{k-1}$ to the L.H.S. above we get

$$a^2 - 2(n-1)\big(1 + 2(n-1)\beta_{k-1}^2 + (n-1)\beta_{k-1}\beta_k\big) + 2(n-1)^2\beta_k\beta_{k-1} - (n-1) \geq 0.$$

By combining this with (B.1) we get

$$a^2 - 2(n-1)a(\beta_k + \beta_{k-1}) + 2(n-1)^2\beta_k\beta_{k-1} - (n-1) \geq 0.$$

By moving some of the terms to the R.H.S. and factorizing the L.H.S. we get

$$\big(a - (n-1)\beta_k\big)\big(a - (n-1)\beta_{k-1}\big) \geq a(n-1)(\beta_k + \beta_{k-1}) - (n-1)^2\beta_k\beta_{k-1} + (n-1),$$

which we can write as

$$1 \geq \frac{(n-1)\big(a(\beta_k + \beta_{k-1}) - (n-1)\beta_k\beta_{k-1} + 1\big)}{\big(a - (n-1)\beta_k\big)\big(a - (n-1)\beta_{k-1}\big)}. \tag{B11}$$

Now, suppose $\beta_k - \beta_{k-1} \geq 0$. Multiplying both sides of the inequality (B.11) by $\beta_k - \beta_{k-1}$, according to (B.6) we get

$$\beta_k - \beta_{k-1} \geq \rho_k - \rho_{k-1},$$

so,

$$\rho_{k-1} - \beta_{k-1} \geq \rho_k - \beta_k$$

which means that via (B.4) we have shown $\beta_k \geq \beta_{k+1}$. Alternatively, if we had $\beta_k - \beta_{k-1} \leq 0$ above, then we would get $\beta_k \leq \beta_{k+1}$. Hence, we always have $(\beta_{k+1} - \beta_k)(\beta_k - \beta_{k-1}) \leq 0$, which is exactly inequality (B.10).

Since we start with $\beta_0 = 1/a$, according to Lemma 2.1 we have $\beta_1 \leq \beta_0$. Using (B.10) inductively we get

$$\beta_1 - \beta_0 \leq 0, 0 \leq \beta_2 - \beta_1, \beta_3 - \beta_2 \leq 0, \ldots$$

and from applying (B.9) to each one of these inequalities we conclude

$$\beta_2 - \beta_0 \leq 0, 0 \leq \beta_3 - \beta_1, \beta_4 - \beta_2 \leq 0, \ldots$$

which shows that we can split $\{\beta_k\}$ into two separate monotonically decreasing and increasing subsequences:

$$0 < \ldots \beta_4 \leq \beta_2 \leq \beta_0 = 1/a,$$
$$0 < \beta_1 \leq \beta_3 \leq \beta_5 \ldots < 1/a.$$

By the bounded monotone convergence theorem we conclude that each one of these subsequences converge, i.e.

$$\lim_{k\to\infty} |\beta_{k+2} - \beta_k| = 0,$$

and recalling (B.5) we get

$$\lim_{k\to\infty} |\rho_{k+1} - \rho_k| = 0.$$

On the other hand, looking at the equality in (B.6) we know that except $(\beta_{k+1} - \beta_k)$ all the factors in the numerator and denominator are bounded away from zero. So therefore we must have

$$\lim_{k\to\infty} |\beta_{k+1} - \beta_k| = 0,$$

and hence, since the even and odd sequences both converge, they must have the same limit. Using the definition of β_{k+1} in (A.1) we get

$$\lim_{k\to\infty} \left| \frac{1 + (n-1)\beta_k^2}{a - (n-1)\beta_k} - 2\beta_k \right| = 0.$$

Since the denominator is bounded away from zero we must have

$$\lim_{k\to\infty} 3(n-1)\beta_k^2 - 2a\beta_k + 1 = 0.$$

The two roots of the limiting quadratic equation are

$$\frac{a \pm \sqrt{a^2 - 3(n-1)}}{3(n-1)}.$$

The smaller root is b as defined in (30) and the larger root is greater than $1/a$, which according to Lemma 2.1 is not possible. Hence,

$$\lim_{k\to\infty} \beta_k = \lim_{k\to\infty} |b_k| = b.$$