Math. Program., Ser. A (2013) 141:135–163
DOI 10.1007/s10107-012-0514-2

FULL LENGTH PAPER

Nonsmooth optimization via quasi-Newton methods

Adrian S. Lewis · Michael L. Overton

Received: 29 August 2010 / Accepted: 3 January 2012 / Published online: 16 February 2012

Abstract We investigate the behavior of quasi-Newton algorithms applied to minimize a nonsmooth function f, not necessarily convex. We introduce an inexact line search that generates a sequence of nested intervals containing a set of points of nonzero measure that satisfy the Armijo and Wolfe conditions if f is absolutely continuous along the line. Furthermore, the line search is guaranteed to terminate if f is semi-algebraic. It seems quite difficult to establish a convergence theorem for quasi-Newton methods applied to such general classes of functions, so we give a careful analysis of a special but illuminating case, the Euclidean norm, in one variable using the inexact line search and in two variables assuming that the line search is exact. In practice, we find that when f is locally Lipschitz and semi-algebraic with bounded sublevel sets, the BFGS (Broyden–Fletcher–Goldfarb–Shanno) method with the inexact line search almost always generates sequences whose cluster points are Clarke stationary and with function values converging R-linearly to a Clarke stationary value. We give references documenting the successful use of BFGS in a variety of nonsmooth applications, particularly the design of low-order controllers for linear dynamical systems. We conclude with a challenging open question.

Keywords BFGS · Nonconvex · Line search · R-linear convergence · Clarke stationary · Partly smooth

Mathematics Subject Classification (2000) 90C30 · 65K05

A. S. Lewis
School of Operations Research and Information Engineering,
Cornell University, Ithaca, NY 14853, USA
e-mail: adrian.lewis@cornell.edu

M. L. Overton (✉)
Courant Institute of Mathematical Sciences, New York University,
New York, NY 10012, USA
e-mail: overton@cs.nyu.edu

1 Introduction

Methods for minimizing functions $f : \mathbf{R}^n \to \mathbf{R}$ which are not differentiable everywhere are based on the observation that the steepest descent (gradient) method routinely fails on such functions, regardless of whether the line search is exact or inexact. By failure we mean that convergence takes place to non-stationary points, as has been known at least since the 1970s and is explained in [19, Section 2.2]. The traditional approach to designing algorithms for nonsmooth optimization is to stabilize steepest descent by exploiting gradient or subgradient information evaluated at multiple points: this is the essential idea of bundle methods [19,22] and also of the gradient sampling algorithm [7,23]. In this paper we investigate the behavior of quasi-Newton (variable metric) methods, particularly the well known BFGS (Broyden–Fletcher–Goldfarb–Shanno) method, when applied to minimize nonsmooth functions, both convex and nonconvex.

It was shown by Powell [42] that, if f is convex and twice continuously differentiable, and the sublevel set $\{x : f(x) \leq f(x_0)\}$ is bounded (x_0 being the starting point), then the sequence of function values generated by the BFGS method with an inexact Armijo–Wolfe line search converges to or terminates at the minimal value of f. This result does not follow directly from the standard Zoutendijk theorem as one needs to know that the eigenvalues of the inverse Hessian approximation H_k do not grow too large or too small. If the convexity assumption is dropped, pathological counterexamples to convergence are known to exist [10,36], but it is widely accepted that the method works well in practice in the smooth, nonconvex case [29]. See [39] for further discussion of quasi-Newton methods for the case that f is smooth.

The behavior of quasi-Newton methods on nonsmooth functions has received little attention. While any locally Lipschitz nonsmooth function f can be viewed as a limit of increasingly ill-conditioned differentiable functions (see [45, Thm 9.67] for one theoretical approach, via "mollifiers"), such a view has no obvious consequence for the algorithm's asymptotic convergence behavior when f is not differentiable at its minimizer. Yet, when applied to a wide variety of nonsmooth, locally Lipschitz functions, not necessarily convex, the BFGS method in particular is very effective, automatically using the gradient difference information to update an inverse Hessian approximation H_k that typically becomes extremely ill-conditioned. As long as the line search never returns a point where f is not differentiable, the method is well defined, and, unlike steepest descent, rarely if ever seems to generate sequences of iterates whose cluster points are not Clarke stationary. As a simple example, let $f(x) = 6|x_1| + 3x_2$. On this function, using a simple bisection-based backtracking line search with Armijo parameter chosen in $[0, \frac{1}{3}]$ and starting at $[2; 3]$, steepest descent generates the sequence $2^{-k}[2(-1)^k; 3]$, $k = 1, 2, \ldots$, converging to the origin. In contrast, BFGS with the same line search rapidly reduces the function value towards $-\infty$ [53]. For functions with bounded sublevel sets, linear (geometric) convergence of the function values to a locally minimal value is typical.

Although there has been little study of this phenomenon in the literature, the frequent success of quasi-Newton methods on nonsmooth functions was observed by Lemaréchal several decades ago. His comments in [27] include:

> We have also exhibited the fact that it can be good practice to use a quasi-Newton method in nonsmooth optimization [as] convergence is rather rapid, and often a reasonably good approximation of the optimum is found; this, in our opinion, is essentially due to the fact that inaccurate line-searches are made. Of course, there is no theoretical possibility to prove convergence to the right point (in fact counterexamples exist), neither are there any means to assess the results.
> ...this raises the question: is there a well-defined frontier between quadratic and piecewise linear, or more generally, between smooth and nonsmooth functions?

For a related discussion, see [19, Ch. VIII, Sec. 3.3].

Lemaréchal's observation was noted in several papers of Lukšan and Vlček [34,35,48]. They wrote in [48]: "standard variable metric methods are relatively robust and efficient even in the nonsmooth case.... On the other hand, no global convergence has been proved for standard variable metric methods applied to nonsmooth problems, and possible failures or inaccurate results can sometimes appear in practical computations". Motivated by the low overhead of quasi-Newton methods, Lukšan and Vlček proposed new methods intended to combine the global convergence properties of bundle methods [19,22] with the efficiency of quasi-Newton methods; Haarala [18] gives a good overview. Other papers that combine ideas from bundle and quasi-Newton methods include [4,33,38,43].

Our interest is in standard quasi-Newton methods, particularly BFGS, with an inexact Armijo–Wolfe line search, applied directly to nonsmooth functions without any modifications. Despite indications to the contrary in the quotes above, the only counterexamples to convergence of which we are aware are either dependent on specialized initial conditions or can be explained by the limitations of rounding errors, and, as we explain later, a simple termination test, similar to that used by bundle methods and the gradient sampling method, can be used to detect approximate Clarke stationarity. Although we are motivated by our successful experience with BFGS as a practical tool for nonsmooth optimization, especially in the nonconvex case, we look closely at one particularly simple convex example: the Euclidean norm $\|\cdot\|$. Our hope is that this will lead the way toward a more complete understanding of the behavior of quasi-Newton methods for general nonsmooth problems.

The paper is organized as follows. We begin with some definitions in Sect. 2. Then, in Sect. 3, we give an analysis of the Broyden class of quasi-Newton methods on the norm function for $n = 2$ when the line search is *exact*. We show that they converge to the origin, spiraling in with a Q-linear rate $\frac{1}{2}$ with respect to the number of line searches, independent of the initial Hessian approximation. Numerical evidence indicates that this property extends to $n > 2$, with a rate of convergence of approximately $1 - 1/\sqrt{2n}$.

The remainder of the paper is devoted to methods using an *inexact* line search. Line searches used by quasi-Newton methods for smooth optimization normally impose an Armijo condition on the function value and a Wolfe condition on the directional derivative. Often, a "strong" version of the Wolfe condition is imposed, insisting on a reduction in the absolute value of the directional derivative, in contrast to the standard condition that requires only an algebraic increase. The latter is all that is required to

ensure positive definiteness of the updated inverse Hessian approximation; nonetheless, it is popular both in textbooks and software to require the "strong" condition, despite the substantial increase in implementation difficulty, perhaps because this is the traditional route to proving convergence results for nonlinear conjugate gradient methods on smooth functions. For nonsmooth optimization, it is clear that enforcing the "strong" Wolfe condition is not possible in general, and it is essential to base the line search on the less restrictive condition. The line search we describe in Sect. 4 is similar to earlier methods in the literature, but our analysis differs. We prove that the line search generates a sequence of nested intervals containing a set of points of nonzero measure that satisfy the Armijo and Wolfe conditions, assuming that f is absolutely continuous along the line. We also prove that the line search terminates under slightly stronger assumptions, in particular covering all semi-algebraic functions (not necessarily locally Lipschitz), and we give a complexity analysis for the case that f is convex. In order to obtain these results we make the idealized assumption that the "oracle" that returns function and gradient values at a given point x is able to detect whether or not f is differentiable along the line at the point x, in contrast to the usual oracle that returns a subgradient instead of a gradient in the nondifferentiable case.

The success of quasi-Newton methods when f is sufficiently smooth with nonsingular Hessian at a minimizer is in large part because inexact line searches quickly find an acceptable step: eventually the method always accepts the unit step and converges superlinearly. The behavior of these methods with an inexact line search on *nonsmooth* functions is complex: it is far from clear whether the direction will be well scaled. As a first analysis of this crucial but difficult question, we carefully consider the univariate case. In Sect. 5 we prove that, for $f(x) = |x|$, the function values computed by a quasi-Newton method converge to zero R-linearly with convergence rate $\frac{1}{2}$. Numerical evidence indicates that this result extends to the norm function with $n > 1$, with a rate of convergence for BFGS of approximately $1 - 1/(2n)$.

In Sect. 6, we summarize our numerical experience with BFGS on nonsmooth functions. We focus on a specific example that illustrates several interesting points: a function defined by a product of eigenvalues. Systematic investigations of other classes of nonsmooth examples appear elsewhere [31]. We have found consistently that, provided the method is initialized randomly, points where f are nondifferentiable are not encountered by the line search and, more surprisingly, cluster points of the algorithm always seem to be Clarke stationary (typically local minimizers). Furthermore, the computed function values converge R-linearly to the Clarke stationary value, with a rate of convergence that varies in an unexpectedly consistent way with the dimension and parameters defining the problem in each class. For some problems, convergence may not be observed, but this seems to be due to rounding error caused by ill-conditioning, not a failure of the method to converge in exact arithmetic. Comparisons with other methods for nonsmooth optimization may be found in [46,47]. A particularly interesting class of examples, Nesterov's nonsmooth Chebyshev-Rosenbrock functions, for which BFGS finds non-minimizing Clarke stationary points, is discussed in [17] and [20]. We give references documenting the successful use of BFGS in several nonsmooth applications, particularly the design of low-order

controllers for linear dynamical systems. We conclude in Sect. 7 with some challenging open questions.

An intuitive, although far from complete, argument for the success of quasi-Newton methods on nonsmooth problems goes as follows. Because the gradient differences may be enormous compared to the difference of the points where they are computed, the inverse Hessian approximation typically becomes very ill-conditioned in the nonsmooth case. Eigenvectors corresponding to tiny eigenvalues of H_k are directions along which, according to the quadratic model constructed by the method, the function has a huge second derivative. In fact, of course, f is not differentiable at the local optimizer being approximated, but can be arbitrarily well approximated by a function with a sufficiently ill-conditioned Hessian. As is familiar from interior-point methods for constrained optimization, it is this ill-conditioning of H_k that apparently enables the method to work so well. Remarkably, if the method is not terminated earlier, it is typical that the condition number of H_k approaches the inverse of the machine precision before rounding errors cause a breakdown in the method, usually failure to obtain a reduction of f in the inexact line search. The spectral decomposition of the final H_k typically reveals two subspaces along which the behavior of f is very different: the eigenvalues that are *not* relatively tiny are associated with eigenvectors that identify directions from the final iterate along which f varies smoothly, while the tiny eigenvalues are associated with eigenvectors along which f varies nonsmoothly. More specifically, when applied to partly smooth functions [28], it seems typical that quasi-Newton methods *automatically* identify the U and V-spaces associated with f at the approximate minimizer. Furthermore, even when H_k is very ill-conditioned, the BFGS direction is typically relatively well scaled, and this property does not deteriorate as the iteration count k increases. Mysteries that remain include the mechanism that prevents the method from stagnating, the reason for the relative well-scaledness of the BFGS direction, and the condition measure of f that determines the surprisingly consistent linear rates of convergence that we observe.

Comments in the literature observing that the popular limited-memory variant of BFGS sometimes works well in practice on nonsmooth problems have appeared occasionally: see [25,54] as well as the comparisons in [47]. Negative comments have also appeared [18, p. 83], [52], leading the authors to propose modifications to the method. Although we have much less experience with the limited-memory variant, we speculate that some of the failures that have been observed may be due to the use of a "strong" Wolfe line search, which can cause failure on simple examples.

2 Definitions

By a *quasi-Newton method* for minimizing a function $f : \mathbf{R}^n \to \mathbf{R}$ we mean the following. Let x_k denote the current point at iteration $k = 0, 1, \ldots$ The gradient of f at x_k is denoted $\nabla f(x_k)$ and abbreviated to ∇f_k. We use H_k to denote a positive definite matrix which is an estimate of the *inverse* Hessian $\nabla^2 f(x_k)^{-1}$.

Algorithm 2.1 (quasi-Newton method)

Choose x_0 with f differentiable at x_0, set H_0 to a positive definite matrix and $k \leftarrow 0$
repeat
set $p_k \leftarrow -H_k \nabla f_k$
set $x_{k+1} \leftarrow x_k + t_k p_k$, where $t_k > 0$ is chosen by a line search
if f is not differentiable at x_{k+1}, or $\nabla f_{k+1} = 0$, stop.
set $y_k \leftarrow \nabla f_{k+1} - \nabla f_k$
choose H_{k+1} to be a positive definite matrix satisfying the secant condition $H_{k+1} y_k = t_k p_k$
$k \leftarrow k + 1$
end (repeat)

If f is not differentiable at x_{k+1} we say that the algorithm *breaks down* (*in theory*). If $\nabla f_{k+1} = 0$ we say it terminates at a *smooth stationary point*. A more practical stopping criterion will be introduced in Sect. 6.

The *BFGS update* is defined by

$$H_{k+1} = V_k H_k V_k^T + t_k (p_k^T y_k)^{-1} p_k p_k^T, \quad \text{where } V_k = I - (p_k^T y_k)^{-1} p_k y_k^T. \tag{2.2}$$

Note that H_{k+1} can be computed in $O(n^2)$ operations since V_k is a rank one perturbation of the identity. There are alternative implementations, notably those that update a factorization of the estimate of $\nabla^2 f(x_k)$ instead of its inverse, but no compelling advantage to these has been established when f is smooth [39].

The *Broyden family* of quasi-Newton updates is defined by a parameter ϕ: when $\phi = 0$, the Broyden update reduces to BFGS, while for $\phi = 1$, it reduces to the Davidon–Fletcher–Powell (DFP) update [39, Sec. 6.3]. The updated matrix H_{k+1} is guaranteed to be positive definite for all $\phi \in [0, 1]$ as long as the line search enforces the Wolfe condition. Powell's result on the convergence of BFGS with an Armijo–Wolfe inexact line search was extended in [8] to the Broyden class for $\phi \in [0, 1)$.

Let A be an invertible $n \times n$ matrix. Applying any method in the Broyden class to the function g defined by $g(x) = f(Ax)$ using starting point x_0 and initial inverse Hessian approximation H_0 is equivalent to replacing g, x_0 and H_0 by f, Ax_0 and AH_0A^T, respectively. This well-known and desirable invariance property of quasi-Newton methods holds regardless of whether f is smooth or not.

When we refer to initializing x and H *randomly*, we mean generating x_0 from the normal distribution and H_0 from the Wishart distribution, that is $H_0 = X^T X$, where the entries of the square matrix X are normally distributed.

We use $\partial f(x)$ to denote the *Clarke subdifferential* [9,45] of f at x, which for locally Lipschitz f is simply the convex hull of the limits of gradients of f evaluated at sequences converging to x [6, Theorem 6.2.5]. An element of $\partial f(x)$ is called a *subgradient* of f at x. A locally Lipschitz, directionally differentiable function f is *regular* at a point when its directional derivative $x \mapsto f'(x; d)$ is upper semicontinuous there for every fixed direction d, and in this case $0 \in \partial f(x)$ is equivalent to the first-order optimality condition $f'(x, d) \geq 0$ for all directions d. Convex functions and smooth functions are regular.

A regular function f is *partly smooth* at x relative to a manifold $\mathcal{M}$ containing x [28] if (1) its restriction to $\mathcal{M}$ is twice continuously differentiable near x, (2) ∂f is continuous on $\mathcal{M}$ near x, and (3) $\operatorname{par} \partial f(x)$, the subspace parallel to the affine hull of the subdifferential of f at x, is exactly the subspace normal to $\mathcal{M}$ at x. For convenience we refer to $\operatorname{par} \partial f(x)$ as the *V-space* for f at x (with respect to $\mathcal{M}$), and to its orthogonal complement, the subspace tangent to $\mathcal{M}$ at x, as the *U-space* for f at x. When we refer to the V-space and U-space without reference to a point x, we mean at a minimizer. For nonzero y in the V-space, the mapping $t \mapsto f(x + ty)$ is necessarily nonsmooth at $t = 0$, while for nonzero y in the U-space, $t \mapsto f(x + ty)$ is differentiable at $t = 0$ as long as f is locally Lipschitz. For example, the Euclidean norm is partly smooth at 0 with respect to the trivial manifold $\{0\}$, the V-space at 0 is $\mathbf{R}^n$, and the U-space is $\{0\}$. When f is convex, the partly smooth nomenclature is consistent with the usage of V-space and U-space in [32]. Most of the functions that we have encountered in applications are partly smooth at local optimizers with respect to some manifold, but many of them are not convex.

The graph of a *semi-algebraic* function is a finite union of sets, each defined by a finite list of polynomial inequalities.

If a sequence $\{\tau_k\}$ converges to a limit μ with $\lim_{k\to\infty} |\tau_{k+1} - \mu|/|\tau_k - \mu| = r$, we say that the convergence of τ_k is *Q-linear* with rate r. If a sequence $\{\upsilon_k\}$ satisfies $|\upsilon_k - \mu| \leq |\tau_k - \mu|$ where $\{\tau_k\}$ converges to μ with Q-linear rate r, then we say that the convergence of υ_k is *R-linear* with rate r.

3 The norm function, with an exact line search

Suppose that the line search in Algorithm 2.1 is *exact*: t_k minimizes the function $t \mapsto f(x_k + tp_k)$. For many nonsmooth functions, the consequence may be that f is not differentiable at x_{k+1}, in which case Algorithm 2.1 breaks down (in theory). The standard approach to nonsmooth optimization allows for the use of a subgradient instead of the gradient at such a point, possibly leading to a null step ($t_k = 0$), but if Algorithm 2.1 is generalized in this way, then using an exact line search it may fail on simple examples [30].

However, such concerns do not apply to the Euclidean norm function $f = \|\cdot\|$, which has only one point where f is not differentiable: the minimizer. We therefore focus our analysis in this section on the norm function.

We first note a well-known property of quasi-Newton methods with an exact line search.

Proposition 3.1 *If the function $t \mapsto f(x_k + tp_k)$ has a local minimizer at t_k and the function f is differentiable at x_{k+1}, then $p_{k+1}^T y_k = 0$.*

Proof The updated matrix H_{k+1} satisfies the secant condition $H_{k+1} y_k = t_k p_k$. The assumptions imply $p_k^T \nabla f_{k+1} = 0$. We deduce

$$y_k^T p_{k+1} = -y_k^T H_{k+1} \nabla f_{k+1} = -t_k p_k^T \nabla f_{k+1} = 0,$$

as required. □

The analysis in the next subsection is limited to two variables, but we will make some experimental observations for $n > 2$ in Sect. 3.2.

3.1 The case $n = 2$

We use the previous result to develop a recursive relationship for the angle defined by the vector x_k, and prove the following result. The quasi-Newton algorithm terminates only if it generates an iterate $x_k = 0$, which can happen only if H_{k-1} is a multiple of the identity, since $\nabla f(x) = \|x\|^{-1}x$.

Theorem 3.2 *Consider Algorithm 2.1 with an exact line search applied to the Euclidean norm in* $\mathbf{R}^2$. *Suppose the algorithm does not terminate. Then the sequence of iterates* $\{x_k\}$ *converges to zero at Q-linear rate* $\frac{1}{2}$, *eventually rotating around zero with consistent orientation, either clockwise or counterclockwise, through an angle of magnitude approaching* $\frac{\pi}{3}$.

Proof For each iteration $k = 0, 1, 2, \ldots$, let θ_k denote the magnitude of the angle between the search direction p_k and the vector $-x_k$. Since the algorithm does not terminate, $\theta_k > 0$. Since H_k is positive definite, p_k is a descent direction: $0 > p_k^T \nabla f_k = -p_k^T x_k$ so $\theta_k < \frac{\pi}{2}$. We seek to express θ_{k+1} in terms of θ_k.

Without loss of generality we can suppose

$$x_k = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } p_k = \begin{bmatrix} -\cos\theta_k \\ \sin\theta_k \end{bmatrix}, \text{ giving } x_{k+1} = \begin{bmatrix} \sin^2\theta_k \\ \sin\theta_k \cos\theta_k \end{bmatrix},$$

since the line search is exact. By Lemma 3.1, the search direction p_{k+1} is orthogonal to the vector

$$y_k = \nabla f_{k+1} - \nabla f_k = \begin{bmatrix} \sin\theta_k \\ \cos\theta_k \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Let ψ denote the magnitude of the angle between y_k and $-x_{k+1}$. Then $\theta_{k+1} = |\frac{\pi}{2} - \psi|$, so

$$\sin\theta_{k+1} = |\cos\psi| = \left|\frac{y_k^T x_{k+1}}{\|y_k\| \cdot \|x_{k+1}\|}\right| = \left|\frac{\sin\theta_k(\sin\theta_k - 1) + \cos^2\theta_k}{\sqrt{(1-\sin\theta_k)^2 + \cos^2\theta_k}}\right|$$
$$= \sqrt{\frac{1-\sin\theta_k}{2}}.$$

Now elementary calculus shows that the mapping $s \mapsto \sqrt{\frac{1-s}{2}}$ maps the interval $[0, 1]$ onto the interval $I = [0, \frac{1}{\sqrt{2}}]$, and is a contraction mapping on I. Hence $\sin\theta_k$ must converge to the fixed point, namely $\frac{1}{2}$, so the angle θ_k approaches $\frac{\pi}{3}$, and the ratio $\frac{\|x_{k+1}\|}{\|x_k\|}$ approaches $\frac{1}{2}$, showing Q-linear convergence.

It remains to show that the orientation of rotation of the iterates x_k is eventually consistent. For large k, we can assume without loss of generality that the iterate x_k

is $[1, 0]^T$, that the next iterate x_{k+1} is close to $[\frac{1}{4}, \frac{\sqrt{3}}{4}]^T$, corresponding to a counterclockwise rotation through an angle of approximately $\frac{\pi}{3}$. Furthermore, the next search direction p_{k+1} is orthogonal to the vector y_k, which is close to $[-\frac{1}{2}, \frac{\sqrt{3}}{2}]^T$. Hence p_{k+1} has the same direction, approximately, as $\pm[\frac{\sqrt{3}}{2}, \frac{1}{2},]^T$, and since it must be a descent direction at x_{k+1}, the sign must be negative. It follows that the next iterate x_{k+2} results from another counterclockwise rotation of approximately $\frac{\pi}{3}$ from x_{k+1}, so the orientation of rotation is indeed eventually consistent.

A more detailed analysis for BFGS [30] shows that the step t_k satisfies

$$t_k \to \frac{1}{4} \text{ as } k \to \infty.$$

Furthermore, the inverse Hessian approximation H_k satisfies

$$\text{spectrum}(H_k) \sim \frac{1}{2^k}\{3 + \sqrt{3}, 3 - \sqrt{3}\}.$$

In fact, it is easy to check directly that the following holds:

Proposition 3.3 (spiral behavior) *Consider Algorithm 2.1 with the BFGS update (2.2), with an exact line search, applied to the Euclidean norm in* $\mathbf{R}^2$*, and initialized by*

$$x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } H_0 = \begin{bmatrix} 3 & -\sqrt{3} \\ -\sqrt{3} & 3 \end{bmatrix}.$$

The method generates a sequence of vectors $\{x_k\}$ *that rotate clockwise through an angle of* $\frac{\pi}{3}$ *and shrink by a factor* $\frac{1}{2}$ *at each iteration.*

3.2 Experiments with $n > 2$

We do not know how to extend the analysis of the previous subsection to $n > 2$. However, numerical experiments implementing the BFGS iteration, or equivalently any method in the Broyden class (see Sect. 3.3), using the easily computed *minimizing* step t_k, indicate that similar results surely hold for $n > 2$. In Fig. 1, the left panel shows the evolution of $f_k = \|x_k\|$ for typical runs for $n = 2, 4, 8$ and 16, with both x and H initialized randomly. The right panel displays estimated Q-linear convergence rates for the sequence $\{f_k\}$ for varying n. Each asterisk plots the mean of 10 observed convergence rates, each computed by a least squares fit to a different randomly initialized sequence. Since the convergence rates are close to 1 for large n, we plot $-\log_2(1-r)$ against $\log_2(n)$, where r is the average estimated convergence rate. The observed rates grow consistently with n, somewhat faster than $1 - 1/\sqrt{2n}$. Furthermore, the rate of convergence is apparently independent of H_0 unless the method terminates at the origin.

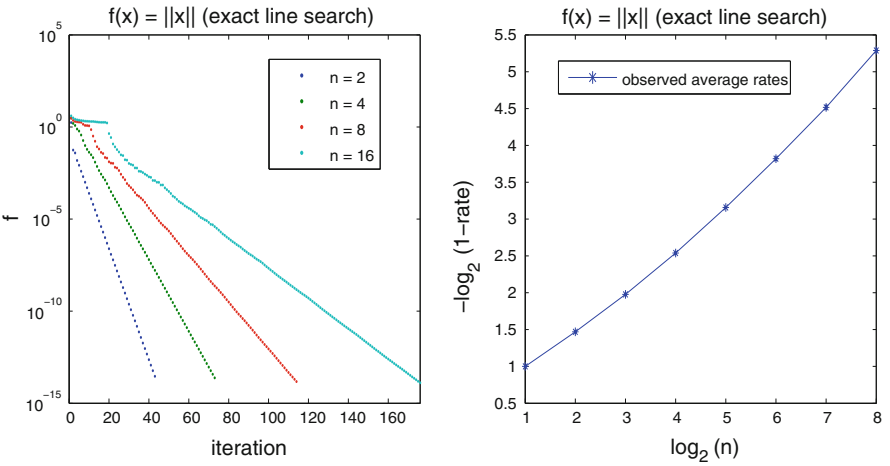

Fig. 1 Convergence of quasi-Newton methods with an exact line search applied to $f(x) = \|x\|$. *Left* plots function values for typical runs for $n = 2, 4, 8$ and 16. *Right* plots $-\log_2(1 - r)$ against $\log_2(n)$ where r is the estimated Q-linear convergence rate for the sequence of function values, averaged over 10 runs

3.3 The Broyden class

Dixon's theorem [13], that all methods in the Broyden family generate the same sequence of iterates $\{x_k\}$ when an exact line search is used, applies to the Euclidean norm function without modification. Thus, the convergence rates in Theorem 3.2 and Fig. 1 apply to the whole Broyden family. However, the steps t_k (and the matrices H_k) *do* depend on the Broyden parameter ϕ.

Numerical experiments on $f = \|\cdot\|$ show that the minimizing steps t_k converge for all $\phi \in [0, 1]$, and Fig. 2 shows their limiting values as a function of ϕ. The left panel shows results for $n = 2$ and the right panel for $n = 16$. Each circle shows the experimentally determined limiting steps, averaged over 10 randomly initialized runs. Experiments were carried out for ϕ ranging from -0.5 to 1.5. When $\phi < 0$, the updated matrix H_k may not be positive definite, and hence t_k may be negative; nonetheless, as long as H_k is never singular, the steps converge to a positive value. For values of ϕ that are sufficiently large, the steps diverge.

The solid curve plots the function $1/(2 - n(\phi - 1))$, which approximates the limiting step well for $n = 2$ and seems to be a reasonably good upper bound when $n > 2$. This implies, in the case $\phi = 0$ (BFGS), that $1/(2 + n)$ is an upper bound on the limiting step. For the case $\phi = 1$ (DFP), the upper bound is $\frac{1}{2}$. The results might suggest that DFP is more favorable for use with an inexact line search as fewer function evaluations would be needed, at least on this example. However, this conclusion overlooks the fact that the limiting step diverges when ϕ is not much greater than 1, specifically somewhat more than the pole in the upper bound formula, $\phi = 1 + 2/n$. This indicates a possible instability for DFP, which is perhaps not surprising, given its well known relatively poor performance, with respect to BFGS, for smooth functions [39].

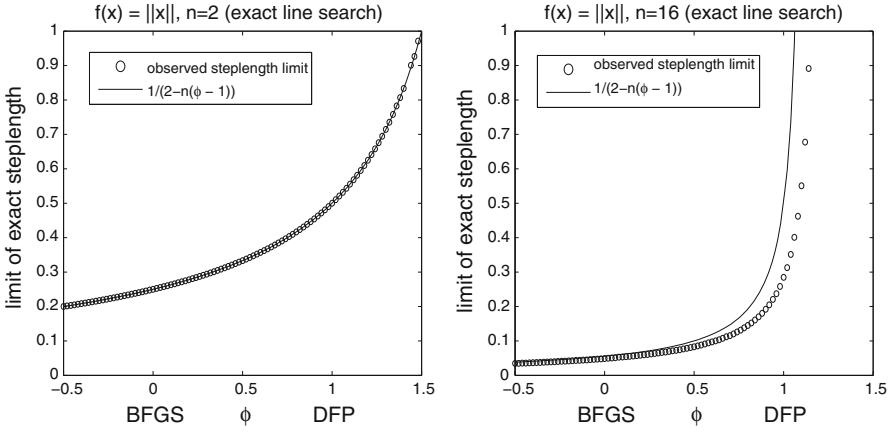

Fig. 2 Limiting steps for the Broyden family using an exact line search on $f(x) = \|x\|$. *Left* the limiting steps as a function of the Broyden parameter ϕ when $n = 2$. *Right* same for $n = 16$

4 An inexact line search for nonsmooth functions

We consider here an inexact line search for nonsmooth optimization very close to one suggested by Lemaréchal [26], and similar to analogous methods of Wolfe [51] (for the convex case) and Mifflin [37]. This line search imposes an Armijo condition on reduction of the function value and a Wolfe condition requiring an algebraic increase in the directional derivative along the line. Our algorithm differs from previous ones in one key respect: how the "oracle" that computes the function and gradient at a given point handles the nondifferentiable case.

Let $\bar{x}$ be an iterate of an optimization algorithm and $\bar{p}$ be a search direction. It is convenient to define the line search objective $h : \mathbf{R}_+ \to \mathbf{R}$ by

$$h(t) = f(\bar{x} + t\bar{p}) - f(\bar{x}).$$

The standard approach to line searches for nonsmooth optimization requires that when f is nondifferentiable at $\bar{x} + \bar{t}\bar{p}$ for a given $\bar{t}$, the oracle computes a subgradient $\bar{g}$ of f at $\bar{x} + \bar{t}\bar{p}$ instead of the gradient, resulting in the subgradient $\bar{g}^T\bar{p}$ of h at $\bar{t}$ instead of $h'(\bar{t})$. In contrast, we assume that the oracle determines whether or not h is differentiable at $\bar{t}$, and if so, it returns $h'(\bar{t})$. This allows us to focus in this section entirely on the properties of the univariate function h without being concerned about the properties of the underlying function f.

We seek a method for selecting a step under the following assumption. If f is differentiable at $\bar{x}$, then the quantity s is $\nabla f(\bar{x})^T\bar{p}$, but we do not need to assume this for the results that follow.

Assumption 4.1 The function $h : \mathbf{R}_+ \to \mathbf{R}$ is absolutely continuous on every bounded interval, and bounded below. Furthermore, it satisfies

$$h(0) = 0 \text{ and } s = \limsup_{t \downarrow 0} \frac{h(t)}{t} < 0.$$

Absolutely continuous functions may be characterized as indefinite integrals of integrable functions [44]. They are differentiable almost everywhere, and satisfy the fundamental theorem of calculus. Lipschitz functions are absolutely continuous, as are semi-algebraic functions. Hence if the function f is locally Lipschitz *or* semi-algebraic, the line search objective h satisfies the absolute continuity assumption.

Given constants $c_1 < c_2$ in the interval $(0, 1)$, we seek an *Armijo–Wolfe step*, which we define to be a number $t > 0$ satisfying the Armijo and Wolfe conditions

$$A(t): \qquad h(t) < c_1 s t \tag{4.2}$$

$$W(t): \qquad h \text{ is differentiable at } t \text{ with } h'(t) > c_2 s. \tag{4.3}$$

Lemma 4.4 *If the condition A holds at the number $\alpha > 0$ but fails at the number $\beta > \alpha$ and the function h is absolutely continuous on the interval $[\alpha, \beta]$, then the set of Armijo–Wolfe steps in the interval $[\alpha, \beta]$ has nonzero measure.*

Proof Since condition A holds at α, by continuity there exists a number γ in the interval $(\alpha, \beta]$ such that A holds throughout the interval $[\alpha, \gamma]$. Now suppose that the conclusion of the lemma fails. Then we must have $h' <= c_2 s$ almost everywhere on the interval $[\alpha, \gamma]$. Thus we can define a number t^* by

$$t^* \;=\; \sup\left\{t \in [\alpha, \beta] : h' \le c_2 s \text{ almost everywhere on } [\alpha, t]\right\}.$$

Then $h' \le c_2 s$ almost everywhere on the interval $[\alpha, t^*]$, so

$$h(t^*) - h(\alpha) = \int_\alpha^{t^*} h' \le c_2 s(t^* - \alpha) \le c_1 s(t^* - \alpha).$$

Since the condition $A(\alpha)$ holds,

$$H(t^*) - c_1 s t^* \le h(\alpha) - c_1 s \alpha < 0,$$

so the condition $A(t^*)$ holds. Since the condition $A(\beta)$ fails, $t^* \ne \beta$, so in fact $t^* < \beta$. By the definition of t^*, for all small $\delta > 0$, condition W must hold on a subset of the interval $[t^*, t^* + \delta]$ of positive measure. But by continuity, the condition A holds throughout this interval for small δ, giving a contradiction. □

Theorem 4.5 (existence of step) *Under Assumption 4.1, the set of Armijo–Wolfe steps has nonzero measure.*

Proof The "lim sup" assumption ensures that there exists $\alpha > 0$ satisfying

$$\frac{h(\alpha)}{\alpha} < c_1 s$$

so condition $A(\alpha)$ holds. On the other hand, condition $A(\beta)$ must fail for all large $\beta > 0$ because the function h is bounded below. Now apply the lemma. □

In fact, for the purposes of the above result, the "lim sup" in Assumption 4.1 could be replaced by "lim inf".

4.1 Definition of the inexact line search

We now define the line search.

Algorithm 4.6 (line search)

$\alpha \leftarrow 0$
$\beta \leftarrow +\infty$
$t \leftarrow 1$
repeat
  if $A(t)$ fails
    $\beta \leftarrow t$
  elseif $W(t)$ fails
    $\alpha \leftarrow t$
  else
    stop
  if $\beta < +\infty$
    $t \leftarrow (\alpha + \beta)/2$
  else
    $t \leftarrow 2\alpha$
end(repeat)

Each execution of the repeat loop involves trying one new choice of the step t, calling the oracle to evaluate $h(t)$ and, when it exists, its derivative $h'(t)$. We call such an execution a *trial*.

Theorem 4.7 (convergence) *Whenever the above line search iteration terminates, the final trial step t is an Armijo–Wolfe step. In particular, it terminates under the assumption*

$$\lim_{t \uparrow \bar{t}} h'(t) \text{ exists in } [-\infty, +\infty] \quad \text{for all } \bar{t} > 0. \tag{4.8}$$

If, on the other hand, the iteration does not terminate, then it eventually generates a nested sequence of finite intervals $[\alpha, \beta]$, halving in length at each iteration, and each containing a set of nonzero measure of Armijo–Wolfe steps. These intervals converge to a step $\tilde{t} > 0$ such that

$$h(\tilde{t}) = c_1 s \tilde{t} \quad \text{and} \quad \limsup_{t \uparrow \tilde{t}} h'(t) \geq c_2 s. \tag{4.9}$$

Proof It is clear that if the line search terminates at t, both conditions $A(t)$ and $W(t)$ hold. Suppose the iteration does not terminate. Eventually, the upper bound β becomes finite, since otherwise condition $A(2^k)$ must hold for all $k = 1, 2, \ldots$, contradicting the boundedness assumption. Furthermore, from the update for β, once β is finite, condition $A(\beta)$ always fails.

Next, notice that eventually the lower bound α is positive. Otherwise, α is always zero, and after the upper bound β becomes finite the trial step t keeps halving and the condition $A(t)$ keeps failing, contradicting the "lim sup" condition in Assumption 4.1. Notice also that after any update to the lower bound α, the condition $A(\alpha)$ must hold.

Let us denote by $[\alpha_k, \beta_k]$ the sequence of intervals generated by the iteration. Once $\alpha_k > 0$ and $\beta_k < \infty$, the intervals are positive, finite, and halve in length at each iteration, and the sequences $\{\alpha_k\}$ and $\{\beta_k\}$ are monotonic increasing and decreasing respectively. Hence there must exist a point $\tilde{t} > 0$ such that $\alpha_k \uparrow \tilde{t}$ and $\beta_k \downarrow \tilde{t}$. Furthermore, we know that the condition $A(\alpha_k)$ holds and the condition $A(\beta_k)$ fails.

We deduce several consequences. First, by the continuity of the function h at the point $\tilde{t}$, we must have $h(\tilde{t}) = c_1 s \tilde{t}$, so the condition $A(\tilde{t})$ fails. On the other hand, the condition $A(\alpha_k)$ holds, so $\alpha_k < \tilde{t}$ for all k. Now, Lemma 4.4 shows the existence of an Armijo–Wolfe step $t_k \in [\alpha_k, \tilde{t}]$. In particular, we know $h'(t_k) > c_2 s$, so property (4.9) follows.

Now suppose assumption (4.8) holds, and yet, by way of contradiction, that the iteration does not terminate but instead generates an infinite sequence of intervals $[\alpha_k, \beta_k]$ as above, shrinking to a point $\tilde{t} > 0$. Every α_k is a trial step at some iteration $j \leq k$, so the condition $W(\alpha_k)$ fails. By our assumption, the function h is differentiable on some nonempty open interval $(t', \tilde{t})$, and hence in particular at α_k for all large k, and so must satisfy $h'(\alpha_k) \leq c_2 s$. We deduce

$$\lim_{t \uparrow \tilde{t}} h'(t) \leq c_2 s < c_1 s. \tag{4.10}$$

On the other hand, h is continuous, so by the Mean Value Theorem there exists a point γ_k in the interval $(\alpha_k, \tilde{t})$ satisfying

$$h'(\gamma_k) = \frac{h(\tilde{t}) - h(\alpha_k)}{\tilde{t} - \alpha_k} \geq \frac{c_1 s \tilde{t} - c_1 s \alpha_k}{\tilde{t} - \alpha_k} = c_1 s.$$

Since γ_k converges to $\tilde{t}$ from the left, this contradicts inequality (4.10). □

The convergence result above is not restricted to Lipschitz functions. In particular, assumption (4.8) holds for any semi-algebraic function h. In contrast with our result, [26] restricts attention to locally Lipschitz functions with a "semismoothness" property. As we now sketch, a very similar argument to the proof above covers that case too.

Suppose the function h is *weakly lower semismooth* at every point $\bar{t} > 0$: in other words, it is locally Lipschitz around $\bar{t}$ and satisfies

$$\liminf_{\tau \downarrow 0} \frac{h(\bar{t} + \tau d) - h(\bar{t})}{\tau} \geq \limsup_k g_k d$$

for $d = \pm 1$ and any sequence of subgradients $\{g_k\}$ of h at $\bar{t} + \tau_k d$ where $\tau_k \downarrow 0$. In the language of [37], this is equivalent to the function $-h$ being "weakly upper semismooth". Suppose in addition that h is differentiable at every trial step. We then claim that the line search terminates.

To see this, assume as in the proof that the iteration does not terminate, so we obtain a sequence of positive numbers $\{\alpha_k\}$ increasing to a point $\tilde{t} > 0$ such that $h(\tilde{t}) = c_1 s\tilde{t}$, the condition $A(\alpha_k)$ holds, the condition $W(\alpha_k)$ fails, and h is differentiable at α_k, for each $k = 1, 2, 3, \ldots$. We deduce the inequalities

$$\begin{aligned}\liminf_{\tau\downarrow 0} \frac{h(\tilde{t}-\tau)-h(\tilde{t})}{\tau} &\le \liminf_k \frac{h(\alpha_k)-h(\tilde{t})}{\tilde{t}-\alpha_k}\\ &\le \liminf_k \frac{c_1 s\alpha_k - c_1 s\tilde{t}}{\tilde{t}-\alpha_k}\\ &= -c_1 s\\ &< -c_2 s\\ &\le \limsup_k h'(\alpha_k)(-1),\end{aligned}$$

which contradicts the definition of weak lower semismoothness.

4.2 Complexity of the line search on a convex function

Unlike the method of [26], due to our different treatment of points where h is not differentiable, our line search method may fail to terminate on some pathological functions, even assuming convexity. For example, consider the function $h\colon \mathbf{R}_+ \to \mathbf{R}$ defined by $h(t) = t^2 - t$ for any number t of the form

$$\eta_k = \sum_{j=0}^{k} (-2)^{-j},$$

and for t equal to 0 or $\frac{2}{3}$ or larger than 1. On the closed intervals between neighboring points of this form, define h by linear interpolation. Then h is convex (although not semi-algebraic), and has a piecewise linear graph with corners $(\eta_k, h(\eta_k))$ accumulating at the point $(\frac{2}{3}, -\frac{2}{9})$. The quantity s defining the Armijo and Wolfe conditions is $h(1/2)/(1/2) = -1/2$, so if $c_1 = \frac{2}{3}$, the points satisfying $A(\cdot)$ constitute the interval $(0, \frac{2}{3})$. For any $c_2 \in (c_1, 1)$, the sequence of trial points is then the sequence of partial sums $\{\eta_k\}$ given above. The condition $A(\eta_k)$ fails for even integers k and holds for odd k, and condition $W(\eta_k)$ always fails due to nondifferentiability. Hence the line search does not terminate.

However, in the convex case we can bound the number of function trials that are needed to generate a point inside an interval in which almost every point satisfies the Armijo and Wolfe conditions.

Proposition 4.11 (complexity of line search) *Consider a convex function h satisfying Assumption 4.1. Then the set of Armijo–Wolfe steps is an open interval $I \subset \mathbf{R}_+$, with any points where h is nondifferentiable removed. Suppose I has left-hand endpoint $b > 0$ and length $a \in (0, +\infty]$. Define*

$$d = \max\{1 + \lfloor \log_2 b \rfloor, 0\}.$$

Then after a number of trials between

$$d+1 \;\; \textit{and} \;\; d+1+\max\left\{d+\left\lfloor \log_2 \frac{1}{a} \right\rfloor, 0\right\}$$

(interpreted in the natural way when $a = +\infty$), the line search tries a step in I.

Proof By convexity, it is easy to see that the interval of interest is given by

$$\begin{aligned} b &= \inf\{t > 0 : W(t) \text{ holds}\} \\ b+a &= \sup\{t > 0 : A(t) \text{ holds}\}. \end{aligned}$$

The line search starts by doubling the search direction until the trial satisfies $t > b$. Assuming this step does not lie in the interval I, the condition $A(t)$ must fail, so the interval $[\alpha, \beta]$ used by the line search to bracket an Armijo–Wolfe step is $[0, t]$. After this doubling phase, the method moves to a bisection phase, repeatedly trying a point t equal to the midpoint of the current bracketing interval. As long as this point lies outside I, the trial t replaces either the left or right endpoint of the bracket, depending on whether $t \le b$ or $t \ge b + a$.

It is easy to see that the number of doublings is d, so the number of trials needed in this phase is $d + 1$. After this phase, the bracketing interval has length 2^d. In fact, if the method continues, the interval I must be contained within the bracket $[2^{d-1}, 2^d]$, and has length a. To find a point in I, the bisection phase repeatedly halves the length of the current bracket. Notice 2^{d-1} is a previous trial point. Hence we need at most

$$\max\left\{d+\left\lfloor \log_2 \frac{1}{a} \right\rfloor, 0\right\}$$

further trials before trying a point in I. The result follows.

In the above result, consider the special case where b is large but $a = 1$, so the interval I is $(b, b + 1)$.

Then the line search will perform a large number,

$$d = 1 + \lfloor \log_2 b \rfloor,$$

of doublings, and then performs between zero and d additional bisections. The point b lies in the interval $[2^{d-1}, 2^d]$. If b lies in the open unit interval $2^d - (0, 1)$, no further trials will be needed. If, on the other hand, b lies in the interval $2^{d-2} + 2^{d-1} - (0, 1)$, one further trial will be needed. Similarly, there exist two open unit intervals of possible values of b requiring two further trials, four requiring three, and more generally, 2^{m-1} unit intervals requiring m further trials, for $m = 1, 2, \ldots, d - 1$. If the point b was a random variable, uniformly distributed in the interval $[2^{d-1}, 2^d]$, the expected number of trials until we try a point in I is then

$$\begin{aligned} &2^{1-d} \cdot 0 + 2^{1-d} \cdot 1 + 2^{2-d} \cdot 2 + 2^{3-d} \cdot 3 + \cdots + 2^{-1} \cdot (d-1) \\ &= 2^{1-d}(1 + 2 \cdot 2 + 2^2 \cdot 3 + \cdots + 2^{d-2} \cdot (d-1)) \\ &= d - 2 + 2^{1-d}. \end{aligned}$$

Thus the expected number of trials in the bisection phase is roughly $\log_2 b$, so the expected total number of trials is about $2 \log_2 b$.

5 The norm function, with the inexact line search

We now consider the behavior of quasi-Newton methods using the line search of Algorithm 4.6 to minimize the Euclidean norm function $\|\cdot\|$. Our analysis in the next subsection is limited to the most trivial case: $n = 1$, but we discuss experimental results for $n > 1$ in Sect. 5.2.

5.1 The absolute value

When $n = 1$, the matrix H_{k+1} is completely defined by the secant equation, so we use the terminology "secant method" instead of quasi-Newton method. Since $f(x) = |x|$, the line search objective is defined by

$$h(t) = |x_k + t p_k| - |x_k|,$$

and Assumption 4.1 is satisfied with

$$p_k x_k < 0 \quad \text{and} \quad s = -|p_k|.$$

Setting the Armijo parameter c_1 to zero simplifies our analysis (we discuss the implications of this choice further below). Since the only point where h is nonsmooth is the minimizer, it also simplifies our analysis to replace the check for differentiability in the Wolfe condition (4.3) by a termination condition. The inequality in (4.3) reduces to $t > -x_k/p_k$ for all $c_2 \in (0, 1)$, so the line search conditions become

$$\begin{aligned} A(t): &\quad t < -2x_k/p_k \\ W(t): &\quad t \geq -x_k/p_k, \end{aligned}$$

with the secant method to be terminated if the line search returns $t_k = -x_k/p_k$. For the analysis that follows, when we refer to the *inexact line search* we mean Algorithm 4.6 with the Armijo and Wolfe conditions redefined as above.

The behavior of the secant method is fundamentally different from that of the steepest descent (gradient) method even on this simple example. In both cases, the iterates converge to zero, but, as we show below, the complexity of the secant method, measured in terms of the *total* number of function trials, is essentially that of a bisection method. In contrast, using the steepest descent method, the search direction is *always* $p_k = \pm 1$, so the closer the iterate x_k is to zero, the more bisections are required to satisfy the Armijo condition in a *single* line search.

Clearly properties A and W guarantee

$$|x_{k+1}| < |x_k| \quad \text{and} \quad x_k x_{k+1} < 0,$$

providing $x_k \neq 0 \neq x_{k+1}$. The inverse Hessian approximation H_{k+1} is defined by the secant equation $H_{k+1} = |x_{k+1} - x_k|/2$ and hence the search direction for the next line search is

$$p_{k+1} = -\frac{|x_{k+1} - x_k|}{2} \operatorname{sgn}(x_{k+1}).$$

Thus, the iterates alternate signs, and the search direction has size half the distance to the previous iterate. This search direction leads to the immediate satisfaction of the Wolfe condition, but one or more bisections may be required until one is found satisfying the Armijo condition (they all satisfy the Wolfe condition).

Now assume for convenience that $H_0 = 1$ and $x_0 \in (\frac{1}{2}, 1)$ so that $x_1 = x_0 - 1$ satisfies both conditions. It is straightforward to check that, with this initialization, the secant method using the inexact line search algorithm on the absolute value function $|\cdot|$ is equivalent to the following algorithm:

Algorithm 5.1

Initialize $x_0 \in (\frac{1}{2}, 1)$ and set $x_1 \leftarrow x_0 - 1, k \leftarrow 1$.
Set $z_0 = x_0, z_1 = x_1$ and $j \leftarrow 1$. Set $w_1 = 1$.

repeat
  $t \leftarrow (x_k + x_{k-1})/2$
  while not done
    $j \leftarrow j + 1$
    $z_j \leftarrow t$ (jth trial point)
    $w_j = |x_k - z_j|$ (current width of interval bracketing zero)
    if $|t| < |x_k|$
      done $\leftarrow$ **true**
    else
      $t \leftarrow (x_k + t)/2$
  end(while)
  $k \leftarrow k + 1$
  $x_k \leftarrow t$ (kth point satisfying Armijo condition)
  if $x_k = 0$, **stop**
end(repeat)

The points $\{x_k\}$ are those where the Armijo condition is satisfied: these are a subsequence of all trial points $\{z_j\}$. Furthermore, it is easy to check that the interval lengths $w_j = |x_k - z_j|$ computed inside the while loop are precisely 2^{1-j}, a sequence converging to zero with Q-linear rate $\frac{1}{2}$. Since x_k and z_j have opposite sign within the while loop, we have $|z_j| < w_j$, and it follows that the sequence of all function trial values $|z_j|$ converges to zero with R-linear rate $\frac{1}{2}$.

A more detailed analysis [31] shows that the process just described is equivalent to computing an "alternating binary expansion" of the initial point x_0. This is summarized in the following result.

Theorem 5.2 *Any number $x \in \mathbf{R}_{++}$ has a unique alternating binary expansion as the sum of a finite or infinite alternating series of strictly decreasing powers of two: that*

is, there is a unique number $m = 0, 1, 2, \ldots$ *or* ∞ *and a unique sequence of integers* $a_0 < a_1 < a_2 < \cdots$ *(with* $m + 1$ *terms if* $m < \infty$*) such that*

$$x = \sum_{j=0}^{m} (-1)^j 2^{-a_j}.$$

Furthermore, applying the secant method to minimize the absolute value function, using the inexact line search, with arbitrary x_0 *and* $H_0 = 1$*, generates the iterates*

$$x_k = \sum_{j=k}^{m} (-1)^j 2^{-a_j} \quad \textit{for all integers } k \leq m. \tag{5.3}$$

Calculating the iterate x_1 *takes* $1 + |a_0|$ *trials in the line search. For all* $k < m$*, given the iterate* x_k*, calculating the subsequent iterate* x_{k+1} *takes* $a_k - a_{k-1}$ *trials. If the alternating binary expansion is finite (that is,* $m < \infty$*), then the secant method terminates at zero after finitely many function trials. Otherwise, the sequence of all function trial values converges to zero with R-linear rate* $\frac{1}{2}$*.*

As an example, consider the initial point

$$x_0 = \frac{4}{7} = 1 - \frac{1}{2} + \frac{1}{8} - \frac{1}{16} + \cdots = \sum_{r=0}^{\infty} (2^{-3r} - 2^{-3r-1})$$

After one trial in the line search, we arrive at the point $x_1 = -3/7$. One more trial takes us to the point $x_2 = 1/14$. The next line search takes two trials before terminating at the point $x_3 = -3/56$. This pattern now repeats: the line search between

$$x_{2j} = \frac{4}{7 \cdot 8^j} \quad \text{and} \quad x_{2j+1} = -\frac{3}{7 \cdot 8^j} \quad \text{for } j = 1, 2, 3, \ldots$$

takes just one trial, but from x_{2j+1} to x_{2j+2} takes two trials. It is easy to confirm that this is exactly the behavior predicted by Theorem 5.2.

Thus, for any initial point $x_0 \in (\frac{1}{2}, 1)$, after a_k trials the secant method guarantees an error less than 2^{-a_k}, and hence the error is reduced to $\epsilon > 0$ after about $\log_2(1/\epsilon)$ trials. By contrast, it is easy to check that steepest descent on $f(x) = |x|$, starting with $x_0 = \frac{2}{3}$, needs $k(k+1)/2$ trials to reduce the error to $2^{1-k}/3$: consequently, reducing the error to ϵ requires about $(\log_2(1/\epsilon))^2/2$ trials.

It is interesting to briefly consider a "tilted" variant of the absolute value function defined by

$$f(x) = \max\{x, -ux\} \quad (x \in \mathbf{R})$$

for a given parameter $u > 0$. When compared with the absolute value, a striking difference emerges: as we let u become large, the Armijo parameter c_1 becomes crucially important. Consider first the case where we apply the secant method with the inexact

line search to f, with the Armijo parameter $c_1 = 0$ as above. Then, an informal analysis and supporting numerical experiments [31] suggest that if the method does not terminate at zero, it generates a sequence of function trial values converging to zero with R-linear rate $r(u)$ satisfying

$$\log_2 r(u) \sim -\frac{1}{\log_2 u} \quad \text{as } u \to +\infty.$$

A very condensed explanation is as follows. Assume that $x_k > 0$. Then, after of the order of $\log_2 u$ trials, the ratio x_{k+2}/x_k may be close to $\frac{1}{2}$, giving a poor convergence rate.

However, restoring the Armijo parameter c_1 to a more standard strictly positive value avoids this slow asymptotic behavior for large u for the following simple reason. The Armijo condition requires

$$x_{k+2} < -ux_{k+1} - c_1 u(x_{k+2} - x_{k+1})$$

from which we deduce

$$x_{k+2} < \frac{1 + c_1}{1 + c_1 u}(-ux_{k+1}) < \frac{1 + c_1}{1 + c_1 u} x_k.$$

Thus, for large u and fixed $c_1 > 0$, the ratio x_{k+2}/x_k has an upper bound behaving like $\frac{1}{u}$.

5.2 Experiments with $n > 1$

It would be interesting to extend the analysis of Sect. 5.1 to the norm function for $n > 1$, but this seems difficult. Numerical experiments indicate, however, that similar results hold. Figure 3 shows the behavior of BFGS with the inexact line search on $f = \|\cdot\|$ when n is varied. The left panel shows *all* function values computed by the algorithm, including trial values in the line search, for typical runs with $n = 1, 2, 4$ and 8. The sequences of function trial values appear to be R-linear: in terms of a semi-log plot such as this, the convergence of a sequence is R-linear with rate $\tilde{r}$ if $\log_{10} \tilde{r}$ is the infimum of the slopes of all lines that bound the points from above. However, our real interest is in the rate of convergence of those function values that are *accepted* by the line search, taking into account nonetheless the number of function evaluations required by the line search: this rate is r if $\log_{10} r$ is the infimum of the slopes of all lines bounding the points corresponding to accepted function values from above. We see from the figure that, for these sequences, the rates $\tilde{r}$ and r are approximately equal. For this reason we estimate the convergence rate of the function trial values using a least squares fit to the pairs (v_k, f_k), where $f_k = \|x_k\|$ is the function value at the end of the kth line search and v_k is the cumulative number of function trials up to that point.

The right panel of Fig. 3 shows the estimated linear convergence rates r computed in this way, averaged over 10 runs, plotting $-\log_2(1 - r)$ against $\log_2(n)$. The

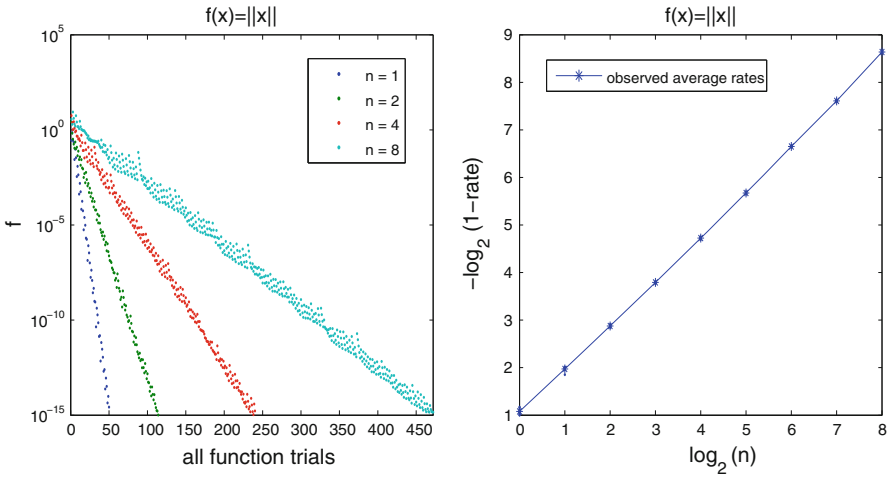

Fig. 3 BFGS with the inexact line search on $f(x) = \|x\|$ for varying n. *Left* typical runs with $n = 1, 2, 4, 8$ showing all function trial values. *Right* plots $-\log_2(1 - r)$, where r is the average observed convergence rate with respect to the number of function trials, against $\log_2(n)$

observed convergence rates are remarkably consistent and we see that r is approximately $1 - 1/(2n)$. It is interesting to compare this to the convergence rate with respect to the number of *exact* line searches for the same problem, which was observed from Fig. 1 to be somewhat greater than $1 - 1/\sqrt{2n}$. The discrepancy between these rates is due to the fact that the average number of function trials needed in an inexact line search grows with n, as can be seen in the left panel of Fig. 3.

For more details on how the experiments were carried out, see the next section.

6 Practical experience

In this section we briefly discuss our practical experience with the BFGS method applied to nonsmooth problems.

6.1 Implementing the inexact line search

For structured functions and initial conditions, the line search might indeed encounter points where h is not differentiable, but in practice this is very unlikely as long as the algorithm is initialized randomly. In any case, in the presence of rounding error, for all but the simplest functions it makes little sense to attempt to check whether either f or h is differentiable at a point, and our line search implementation is based on the assumption that f and therefore also h is differentiable at every point where it is evaluated. For the same reason, while in principle traditional methods for nonsmooth optimization compute subgradients instead of gradients in the nondifferentiable case, in practice they almost always return gradients. Thus, despite the theoretical difference between

our line search and more traditional ones, there is virtually no practical difference, and the result is that our line search behaves like the one in [26] in practice.

If the line search is unable to satisfy the Armijo and Wolfe conditions within a prescribed number of trials, or if the computed value $h'(0) = \nabla f(x_k)^T p_k$ is nonnegative, we say that Algorithm 2.1 *breaks down* (*in practice*). Although in principle such breakdown might occur because f is not differentiable at x_k, in practice breakdown seems to simply be a consequence of the limitations of machine precision.

The results reported here use the value zero for the Armijo parameter c_1, but they are essentially the same when c_1 is set to a small positive value. We used the value $1/2$ for the Wolfe parameter c_2.

6.2 An example: minimizing a product of eigenvalues

We have found that the BFGS algorithm with the inexact line search converges consistently to Clarke stationary points (usually, local minimizers) on many different kinds of examples [31]. Here we present results for one illustrative example: an entropy minimization problem arising in an environmental application [2]. Let $\mathcal{S}^N$ denote the space of real symmetric N by N matrices. The function f to be minimized is

$$f(X) = \log E_K(A \circ X),$$

where $E_K(X)$ denotes the product of the K largest eigenvalues of a matrix X in $\mathcal{S}^N$, A is a fixed matrix in $\mathcal{S}^N$, and $\circ$ denotes the Hadamard (componentwise) matrix product, subject to the constraints that X is positive semidefinite and has diagonal entries equal to 1. If the requirement were to minimize the *sum* of the largest eigenvalues instead of the product, this would be equivalent to a semidefinite program, but the product of the largest K eigenvalues is not convex. This problem was one of the examples in [7]; in the results reported there, the objective function was defined without the logarithm and we enforced the semidefinite constraint by an exact penalty function. Here, we impose the constraint by the substitution $X = VV^T$, where V is square. The constraint on the diagonal of X then translates to a requirement that the rows of V have norm one, a constraint that can be removed from the problem by replacing each row v of V by $v/\|v\|$. Thus, the problem is converted to the unconstrained minimization of a nonsmooth function f over $\mathbf{R}^n$ with $n = N^2$ (the variable being $x = \mathrm{vec}(V)$, the vector representation of the matrix V). In principle, one might expect multiple local minimizers with different minimal values, but at least with the data we have been using, this rarely happens.

Let $\lambda_i(Y)$ denote the ith largest eigenvalue of $Y \in \mathcal{S}^N$ and, for given Y, define an active set $I(Y) = \{i : \lambda_i(Y) = \lambda_K(Y)\}$. It can be verified that E_K is partly smooth at Y with respect to the manifold $\widetilde{\mathcal{M}}(Y) = \{Z \in \mathcal{S}^N : \lambda_i(Z) = \lambda_K(Z) \iff i \in I(Y)\}$. It is known from matrix theory that the codimension of $\widetilde{\mathcal{M}}(Y)$ is $m(m+1)/2 - 1$, where m is the multiplicity $|I(Y)|$ [24, p. 141]. Now consider the manifold in $\mathbf{R}^n$ defined by

$$\mathcal{M}(\bar{x}) = \left\{ x : A \circ \mathrm{vec}(x)\mathrm{vec}(x)^T \in \widetilde{\mathcal{M}}\left(A \circ \mathrm{vec}(\bar{x})\mathrm{vec}(\bar{x})^T\right)\right\},$$

where $\bar{x}$ is a minimizer of f. To conclude that f is partly smooth with respect to $\mathcal{M}$ at $\bar{x}$, and that the codimension of $\mathcal{M}$ is $m(m+1)/2-1$, where $m = |I(A \circ \mathrm{vec}(\bar{x})\mathrm{vec}(\bar{x})^T)|$, requires a transversality condition [28]; let us assume that this holds. For the results reported below, A is set to the leading $N \times N$ submatrix of a 63×63 covariance matrix [2], scaled so that the largest entry is 1, with $N = 20$ ($n = 400$) and $K = 10$.

Figure 4 shows results obtained by running BFGS with the inexact line search 10 different times, each with both x and H initialized randomly, and with each run terminated when the algorithm breaks down (in practice). All 10 runs generated the same final value of f to about 14 digits (-4.3793775559927), with the function trial values converging R-linearly at a consistent rate. Repeated experiments with other problem variants and other nonsmooth optimization methods indicate that this value is, almost certainly, a locally minimal value, although all we can conclude from an *a posteriori* analysis (see the stopping criterion in the next section) is that the final value of x is approximately Clarke stationary. At the top left of Fig. 4, the values of f after each line search are plotted, shifted by f_{opt}, an estimate of the optimal value defined to be the best value found in these 10 runs; the apparent superlinear convergence of f to the optimal value in one run is an artifact of this choice. At the top right, we see the eigenvalues of $A \circ X$ as a function of the iteration count. Observe that after just a few iterations, $\lambda_6(A \circ X), \ldots, \lambda_{14}(A \circ X)$ have coalesced together to plotting accuracy (λ_{15}, λ_{16} and λ_{17} are slightly smaller). This computed multiplicity–9 eigenvalue suggests that the manifold $\mathcal{M}(\bar{x})$ has codimension $9(10)/2 - 1 = 44$; if so, this is the dimension of the V-space at $\bar{x}$. Indeed, this is confirmed by the bottom left plot: exactly 44 eigenvalues of the inverse Hessian approximation matrix H_k converge to zero! Furthermore, at the bottom right we see the function $f - f_{\mathrm{opt}}$ plotted along lines through the computed minimizer x_{opt} parallel to the eigenvectors corresponding to the jth *smallest* eigenvalue of the final computed H, for $j = 10, 20, \ldots, 60$. We see that f is V-shaped in the first four of these directions and U-shaped in the last two, again consistent with our conclusion that the V-space has dimension 44. This is compelling evidence that BFGS *automatically* identifies the U and V-spaces at the local minimizer, without any *a priori* information about the manifold $\mathcal{M}$.

Most important of all is the observation that, regardless of the initial conditions, BFGS generates sequences of function values that converge to Clarke stationary values and with final iterates x which are extremely close to points where f is not differentiable. Indeed, all 10 runs produce a final point x for which $A \circ X$ has an eigenvalue with multiplicity 9 to about 14 digits (nearly the full precision of 16 digits carried by IEEE floating point arithmetic). Steepest descent generates sequences of function values for which the final iterates x are also very close to points where f is not differentiable, but neither the final function values, nor the multiplicity of the eigenvalues of the final $A \circ X$, agree from one run to another, indicating that, as mentioned in Sect. 1, steepest descent routinely generates sequences that converge to points at which f is not differentiable but which are not Clarke stationary.

For an example for which BFGS finds Clarke stationary points that are *not* necessarily local minimizers, see [17,20].

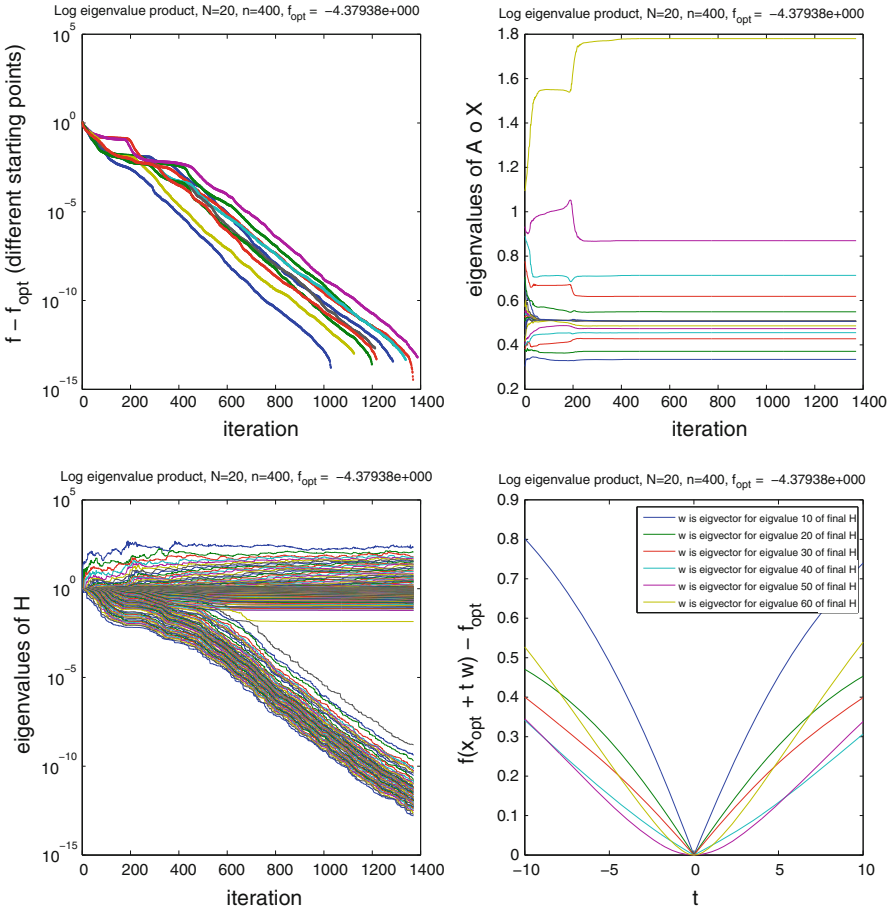

Fig. 4 Results for minimizing the eigenvalue product, $N = 20, n = 400, K = 10$. *Top left* the function values after each line search for 10 randomly generated starting points, shifted by f_{opt}, the minimal value found. *Top right* eigenvalues of $A \circ X$ after each line search for one run. *Bottom left* eigenvalues of H_k for same run: 44 of these converge to zero. *Bottom right* plots $f - f_{\text{opt}}$ along a line $x_{\text{opt}} + t\,w$, where x_{opt} is the computed minimizer and w is the eigenvector of the final H associated with its jth smallest eigenvalue, for $j = 10, 20, \ldots, 60$. The function f is "V-shaped" along the eigenvectors associated with tiny eigenvalues of H, and "U-shaped" along the others

6.3 A stopping condition

It might be thought that a disadvantage of using a quasi-Newton method for nonsmooth optimization is that there is no obvious way to decide how to terminate the method: ill-conditioning of H_k proves nothing and computing the eigenvalues or condition number of H_k would add far too much computational overhead to the iteration. However, the following simple approach can be used to detect approximate Clarke stationarity. Let J be a positive integer and let τ_x and τ_d be two small positive tolerances, all specified by the user or given default values. Define $j_0 = 1$ and $G_0 = \{\nabla f_0\}$ and, for $k = 1, 2, \ldots$, define

$$j_k = 1,\ G_k = \{\nabla f_k\} \quad \text{if} \quad \|x_k - x_{k-1}\| > \tau_x,$$
$$j_k = j_{k-1} + 1,\ G_k = \{\nabla f_{k-j_k+1}, \ldots, \nabla f_k\} \quad \text{if} \quad \|x_k - x_{k-1}\| \le \tau_x \text{ and } j_{k-1} < J,$$
$$j_k = J,\ G_k = \{\nabla f_{k-J+1}, \ldots, \nabla f_k\} \quad \text{if} \quad \|x_k - x_{k-1}\| \le \tau_x \text{ and } j_{k-1} = J.$$

By construction, G_k is a set of $j_k \le J$ gradients evaluated at points near x_k. The smallest vector in the convex hull of this set,

$$d_k = \arg\min\{\|d\| : d \in \operatorname{conv} G_k\},$$

is obtained (as in bundle methods) by solving a convex quadratic program in j_k variables, an inexpensive computation if j_k is small and in any case one whose cost can be reduced by exploiting the information available from iteration $k-1$. Algorithm 2.1 may then be terminated if $\|d_k\| \le \tau_d$, as this inequality is an approximate Clarke stationarity condition when τ_x and τ_d are small. Note that if $J = 1$, the test reduces to $\|\nabla f_k\| \le \tau_d$, the usual stopping condition in practice when f is smooth.

Suppose f is partly smooth at a Clarke stationary point to which the iteration converges. If J is larger than the dimension of the V-space at the minimizer, we typically find that the termination condition just described is satisfied eventually as long as τ_x and τ_d are not so small that breakdown (in practice) occurs first. Appropriate choices for J, τ_x and τ_d are problem dependent. For example, consider the eigenvalue product example of Sect. 6.2, with $n = 400$, for which we argued that the dimension of the V-space at the minimizer found by BFGS is 44. Using $J = 50$ and $\tau_x = \tau_d = 10^{-4}$, BFGS typically terminates successfully in 600–1,000 iterations.

6.4 Software

Our MATLAB package HANSO (Hybrid Algorithm for Non-Smooth Optimization) is based on BFGS and freely available.[1] Version 2.0 of HANSO uses the stopping criterion just described. If the algorithm breaks down (in practice) without satisfying the desired termination condition, the user has the option to continue the optimization using the gradient sampling method of [7]. The gradient sampling method is far more computationally intensive than BFGS, but it does enjoy convergence guarantees with probability one [7,23].

Our BFGS implementation in HANSO has been used to solve a variety of practical nonsmooth problems, such as a condition geodesic problem [3] and shape optimization for spectral functions of Dirichlet-Laplacian operators [40].

Together with D. Henrion, M. Millstone and S. Gumussoy, we have also developed a more specialized package HIFOO (H-Infinity Fixed-Order Optimization) [5,14], also freely available.[2] Its purpose is to design low-order feedback controllers for linear dynamical systems. HIFOO sets up certain small-dimensional but challenging nonsmooth, nonconvex optimization problems and then solves them by calling HANSO.

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

[2] http://www.cs.nyu.edu/overton/software/hifoo/.

The effectiveness of HIFOO in designing low-order controllers is benchmarked in [1,14–16]. Recently published applications of HIFOO include design of teleoperations for minimally invasive surgery [11], design of an aircraft nose landing gear steering system [41], design of an aircraft controller for improved gust alleviation and passenger comfort [50], robust controller design for a proton exchange membrane fuel cell system[49], design of power systems controllers [12] and design of winding systems for elastic web materials [21].

7 A challenge

This paper raises far more questions than it answers. We hope that we have made a convincing case that quasi-Newton methods are practical and effective methods for nonsmooth optimization, and we have tried to give insight into *why* they work so well, but a general analysis seems to be difficult.

In our experience with functions with bounded sublevel sets, BFGS essentially always generates function values converging linearly to a Clarke stationary value, with exceptions only in cases that we attribute to the limits of machine precision. We speculate that, for some broad class of reasonably well-behaved functions, this behavior is almost sure. In framing our challenge, let us first rule out the worst kinds of pathology by considering objective functions whose graphs stratify into analytic manifolds. (A variety of dynamical systems associated with such functions are known to behave well.) To be concrete, we restrict our attention to the class of semi-algebraic functions. Now let us consider appropriately random initial data: the precise distributions are irrelevant, providing they are absolutely continuous with respect to Lebesgue measure. Again to be concrete, let us assume a normally distributed initial point and an initial positive definite inverse Hessian approximation sampled from a Wishart distribution. We now consider the BFGS method, in exact arithmetic, using the inexact line search with any fixed Armijo and Wolfe parameters satisfying $0 < c_1 < c_2 < 1$. Theorem 4.7 guarantees that the line search must always terminate because of the semi-algebraic assumption, but it does not guarantee that f is differentiable at the new iterate x_{k+1} (only that its derivative along the previous direction p_k exists).

Challenge 7.1 *Let f be locally Lipschitz and semi-algebraic with bounded sublevel sets. Prove or disprove that, if x_0 and H_0 are chosen randomly as just described, then the following propositions hold with probability one:*

1. *Algorithm 2.1 using the BFGS update (2.2) and the line search of Algorithm 4.6 does not break down (in theory) and does not terminate at a smooth stationary point.*
2. *Any cluster point $\bar{x}$ of the sequence $\{x_k\}$ is Clarke stationary, that is $0 \in \partial f(\bar{x})$.*
3. *The sequence of all function trial values converges to $f(\bar{x})$ R-linearly.*
4. *Let W_k be the subspace spanned by the eigenvectors associated with the eigenvalues of H_k that converge to zero, and suppose that x_k converges to a point $\bar{x}$ where f is partly smooth with respect to a manifold $\mathcal{M}$. Then W_k converges to the V-space of f at $\bar{x}$ with respect to $\mathcal{M}$, or equivalently, its orthogonal complement converges to the U-space, that is the tangent space to $\mathcal{M}$ at $\bar{x}$.*

Acknowledgments We thank a referee of earlier versions of this paper for suggesting several improvements, including the proof of Theorem 3.2 and the argument for R-linear convergence based on Algorithm 5.1, both of which simplified our original arguments. The same referee pointed out the importance of the Armijo parameter in the context discussed at the end of Sect. 5.1. We are also grateful to another referee for reading the paper carefully and making many useful suggestions. We thank K. Anstreicher and J. Lee for suggesting the eigenvalue product problem of Sect. 6.2. Finally, we thank F. Facchinei for providing a stimulating environment at Università di Roma "La Sapienza", where much of our original work on this paper was completed. Adrian S. Lewis is supported in part by National Science Foundation Grant DMS-0806057. Michael L. Overton is supported in part by National Science Foundation Grant DMS-1016325.

References

1. Arzelier, D., Gryazina, E.N., Peaucelle, D., Polyak, B.T.: Mixed LMI/randomized methods for static output feedback control design. Technical report 09535, LAAS-CNRS, Toulouse, Sept 2009
2. Anstreicher, K., Lee, J.: A masked spectral bound for maximum-entropy sampling. In: di Bucchianico, A., Läuter, H., Wynn eds, H.P. (eds.) MODA 7—Advances in Model-Oriented Design and Analysis, pp. 1–10. Springer, Berlin (2004)
3. Boito, P., Dedieu, J.-P.: The condition metric in the space of rectangular full rank matrices. SIAM J. Matrix Anal. Appl. **31**, 2580–2602 (2010)
4. Bonnans, J., Gilbert, J., Lemaréchal, C., Sagastizábal, C.: A family of variable metric proximal methods. Math. Program. **68**, 15–48 (1995)
5. Burke, J.V., Henrion, D., Lewis, A.S., Overton, M.L.: HIFOO—a MATLAB package for fixed-order controller design and H_∞ optimization. In: Proceedings of Fifth IFAC Symposium on Robust Control Design, Toulouse (2006)
6. Borwein, J.M., Lewis, A.S.: Convex Analysis and Nonlinear Optimization: Theory and Examples. 2nd edn. Springer, New York (2005)
7. Burke, J.V., Lewis, A.S., Overton, M.L.: A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM J. Optim. **15**, 751–779 (2005)
8. Byrd, R.H., Nocedal, J., Yuan, Y.: Global convergence of a class of quasi-Newton methods on convex problems. SIAM J. Numer. Anal. **24**, 1171–1190 (1987)
9. Clarke, F.H.: Optimization and Nonsmooth Analysis. John Wiley, New York, 1983. Reprinted by SIAM, Philadelphia (1990)
10. Dai, Y.-H.: Convergence properties of the BFGS algorithm. SIAM J. Optim. **13**, 693–701 (2002)
11. Delwiche, T.: Contribution to the design of control laws for bilateral teleoperation with a view to applications in minimally invasive surgery. Ph.D. thesis, Free University of Brussels (2009)
12. Dotta, D., e Silva, A.S., Decker, I.C.: Design of power systems controllers by nonsmooth, nonconvex optimization. In: IEEE Power and Energy Society General Meeting, Calgary (2009)
13. Dixon L.C.W.: Quasi-Newton techniques generate identical points. II. The proofs of four new theorems. Math. Program. **3**, 345–358 (1972)
14. Gumussoy, S., Henrion, D., Millstone, M., Overton, M.L.: Multiobjective robust control with HIFOO 2.0. In: Proceedings of the Sixth IFAC Symposium on Robust Control Design, Haifa (2009)
15. Gumussoy, S., Millstone, M., Overton, M.L.: H-infinity strong stabilization via HIFOO, a package for fixed-order controller design. In: Proceedings of the 47th IEEE Conference on Decision and Control, Cancun (2008)
16. Gumussoy, S., Overton, M.L.: Fixed-order H-infinity controller design via HIFOO, a specialized nonsmooth optimization package. In: Proceedings of 2008 American Control Conference, Seattle (2008)
17. Gürbüzbalaban, M., Overton, M.L.: On Nesterov's nonsmooth Chebyshev–Rosenbrock functions. Nonlinear Anal. Theory Methods Appl. **75**, 1282–1289 (2012)
18. Haarala, M.: Large-scale nonsmooth optimization: variable metric bundle method with limited memory. Ph.D. thesis, University of Jyväskylä, Finland (2004)
19. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms, two volumes. Springer, New York (1993)
20. Kaku, A.: Implementation of high precision arithmetic in the BFGS method for nonsmooth optimization. Master's thesis, New York University, Jan 2011. http://www.cs.nyu.edu/overton/mstheses/kaku/msthesis.pdf

21. Knittel, D., Henrion, D., Millstone, M., Vedrines, M.: Fixed-order and structure H-infinity control with model based feedforward for elastic web winding systems. In: Proceedings of the IFAC/IFORS/IMACS/IFIP Symposium on Large Scale Systems, Gdansk, Poland (2007)
22. Kiwiel, K.C.: Methods of descent for nondifferentiable optimization. In: Lecture Notes in Mathematics, vol. 1133. Springer, Berlin (1985)
23. Kiwiel, K.C.: Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM J. Optim. **18**, 379–388 (2007)
24. Lax, P.D.: Linear Algebra. Wiley, New York (1997)
25. Lee, J.: Constrained maximum-entropy sampling. Oper. Res. **46**, 655–664 (1998)
26. Lemaréchal, C.: A view of line searches. In: Optimization and optimal control (Proceedings of Conference at the Mathematical Research Institute, Oberwolfach, 1980), pp. 59–78. Springer, Berlin/New York, 1981. Lecture Notes in Control and Information Sciences, vol. 30
27. Lemaréchal, C.: Numerical experiments in nonsmooth optimization. In: Nurminski, E.A. (ed.) Progress in Nondifferentiable Optimization, pp. 61–84. International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria (1982)
28. Lewis, A.S.: Active sets, nonsmoothness and sensitivity. SIAM J. Optim. **13**, 702–725 (2003)
29. Li, D.-H., Fukushima, M.: On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. SIAM J. Optim. **11**, 1054–1064 (2001)
30. Lewis, A.S., Overton, M.L.: Behavior of BFGS with an exact line search on nonsmooth examples. http://www.cs.nyu.edu/overton/papers/pdffiles/bfgs_exactLS.pdf (2008)
31. Lewis, A.S., Overton, M.L.: Nonsmooth optimization via BFGS. http://www.cs.nyu.edu/overton/papers/pdffiles/bfgs_inexactLS.pdf (2008)
32. Lemaréchal, C., Oustry, F., Sagastizábal, C.: The U-Lagrangian of a convex function. Trans. Am. Math. Soc. **352**, 711–729 (2000)
33. Lemaréchal, C., Sagastizábal, C.: An approach to variable metric bundle methods. In: IFIP Proceedings, Systems Modeling and Optimization (1994)
34. Lukšan, L., Vlček, J.: Globally convergent variable metric method for convex nonsmooth unconstrained minimization. J. Optim. Theory Appl. **102**, 593–613 (1999)
35. Lukšan, L., Vlček, J.: Variable metric methods for nonsmooth optimization. Technical report 837, Academy of Sciences of the Czech Republic, May 2001
36. Mascarenhas, W.F.: The BFGS method with exact line searches fails for non-convex objective functions. Math. Program. **99**, 49–61 (2004)
37. Mifflin, R.: An algorithm for constrained optimization with semismooth functions. Math. Oper. Res. **2**, 191–207 (1977)
38. Mifflin, R., Sun, D., Qi, L.: Quasi-Newton bundle-type methods for nondifferentiable convex optimization. SIAM J. Optim. **8**, 583–603 (1998)
39. Nocedal, J., Wright, S.J.: Nonlinear Optimization. 2nd edn. Springer, New York (2006)
40. Osting, B.: Optimization of spectral functions of Dirichlet-Laplacian eigenvalues. J. Comput. Phys. **229**, 8578–8590 (2010)
41. Pouly, G., Lauffenburger, J.-P., Basset, M.: Reduced order H-infinity control design of a nose landing gear steering system. In: Proceedings of 12th IFAC Symposium on Control in Transportation Systems (2010)
42. Powell, M.J.D.: Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In: Nonlinear Programming, pp. 53–72. American Mathematical Society, Providence. SIAM-AMS Proceedings, vol. IX (1976)
43. Rauf, A.I., Fukushima, M.: Globally convergent BFGS method for nonsmooth convex optimization. J. Optim. Theory Appl. **104**, 539–558 (2000)
44. Royden, H.L.: Real Analysis. Macmillan, New York (1963)
45. Rockafellar, R.T., Wets, R.J.B.: Variational Analysis. Springer, New York (1998)
46. Sagastizábal, C.: Composite proximal bundle method. Technical report. http://www.optimization-online.org/DB_HTML/2009/07/2356.html (2009)
47. Skajaa, A.: Limited memory BFGS for nonsmooth optimization. Master's thesis, New York University, Jan 2010. http://www.cs.nyu.edu/overton/mstheses/skajaa/msthesis.pdf
48. Vlček, J., Lukšan, L.: Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. J. Optim. Theory Appl. **111**, 407–430 (2001)
49. Wang, F.-C., Chen, H.-T.: Design and implementation of fixed-order robust controllers for a proton exchange membrane fuel cell system. Int. J. Hydrogen Energy, **34**, 2705–2717 (2009)

50. Wildschek, A., Maier, R., Hromcik, M., Hanis, T., Schirrer, A., Kozek, M., Westermayer, C., Hemedi M.: Hybrid controller for gust load alleviation and ride comfort improvement using direct lift control flaps. In: Proceedings of Third European Conference for Aerospace Sciences (EUCASS) (2009)
51. Wolfe, P.: A method of conjugate subgradients for minimizing nondifferentiable functions. Math. Program. Stud. **3**, 145–173, (1975). In: Balinski, M.L., Wolfe, P. (eds.) Nondifferentiable Optimization
52. Yu, J., Vishwanathan, S.V.N., Günther, S., Schraudolph, N.: A quasi-Newton approach to non-smooth convex optimization. In: Proceedings of the 25th International Conference on Machine Learning (2008)
53. Zhang, S.S.: Cornell University, Private Communication (2010)
54. Zhang, S., Zou, X., Ahlquist, J., Navon, I.M., Sela, J.G.: Use of differentiable and nondifferentiable optimization algorithms for variational data assimilation with discontinuous cost functions. Mon. Weather Rev. **128**, 4031–4044 (2000)