J Optim Theory Appl (2015) 165:151–171
DOI 10.1007/s10957-014-0622-7

Nonsmoothness and a Variable Metric Method

A. S. Lewis · S. Zhang

Received: 29 June 2014 / Accepted: 4 July 2014 / Published online: 7 August 2014
© Springer Science+Business Media New York 2014

Abstract This paper investigates the behavior, both good and bad, of the Broyden–Fletcher–Goldfarb–Shanno algorithm for smooth minimization, when applied to nonsmooth functions. We consider three particular examples. We first present a simple nonsmooth example, illustrating how this variable metric method (in this case with an exact line search) typically succeeds despite nonsmoothness. We then study, computationally, the behavior of the method with an inexact line search on the same example and discuss the results. In further support of the heuristic effectiveness of the method despite nonsmoothness, we prove that, for the very simplest class of nonsmooth functions (maximums of two affine functions), the method cannot stall at a nonstationary point. On the other hand, we present a nonsmooth example where the method with an inexact line-search converges to a stationary point notwithstanding the presence of directions of linear descent. Finally, we briefly compare line-search and trust-region strategies for this method in the nonsmooth case.

Keywords Broyden–Fletcher–Goldfarb–Shanno (BFGS) · Nonsmooth · Line search · Partial smoothness · Stationary point · Trust region

Mathematics Subject Classification 90C30 · 65K05

Communicated by Asen L. Donchev.

A. S. Lewis · S. Zhang (✉)
ORIE, Cornell University, Ithaca, NY 14853, USA
e-mail: sz254@cornell.edu

A. S. Lewis
e-mail: adrian.lewis@cornell.edu

1 Introduction

We study the behavior of the standard Broyden–Fletcher–Goldfarb–Shanno (BFGS) variable metric method for smooth minimization, when applied to nonsmooth functions. The theory for BFGS applied to convex smooth functions is well established: Powell [1] showed that BFGS with a Wolfe inexact line search converges to a minimizer, when applied to a twice-differentiable convex function with bounded level sets. However, there is no corresponding convergence result for nonconvex smooth functions. Although various authors [2–5] made some progress by restricting the function class or modifying the method, the convergence theory for the BFGS algorithm on nonconvex functions remains poorly understood: see [6,7]. There is even less experience with the BFGS method on nonsmooth functions. The success of variable metric methods on nonsmooth functions was observed many years ago [8], but it seems very challenging to give any rigorous convergence analysis.

Recent work by Lewis and Overton [9] gives a detailed analysis of the BFGS method with an exact line search on one particular example: the Euclidean norm function in $\mathbb{R}^2$. While special, the analysis illustrates how the BFGS method can work well on nonsmooth functions. That paper also investigates the behavior of BFGS with a suitable inexact line search on some nonsmooth examples: the authors observe that this inexact-line-search BFGS method typically converges to Clarke stationary points, and they pose the following challenge, to prove or disprove.

> Consider any locally Lipschitz, semi-algebraic function f with bounded level sets, and choose the initial point x_0 and initial inverse Hessian estimate H_0 randomly. With probability one, the BFGS method generates an infinite sequence of iterates, for which any cluster point $\bar{x}$ is Clarke stationary, and, furthermore, the sequence of all function trial values converges to $f(\bar{x})$ R-linearly.

For more precise details on the terminology, see [9, Challenge 7.1].

Our current work is largely motivated by this earlier paper. We highlight further the success of the line-search BFGS method on some nonsmooth examples and analyze the potential reasons. By way of contrast, we illustrate the potential bad behavior of the line-search BFGS method by constructing a nonsmooth function on which the method converges to a point at which there exist directions of linear descent. Our goal, throughout, is simply insight into the line-search BFGS method in the nonsmooth case.

As context, we also briefly discuss the behavior of a trust-region BFGS method, when applied to nonsmooth functions. The line search and trust-region philosophies for updating the current point of course differ considerably: trust-region methods [10] approximate the original problem in a "trust region" by a quadratic subproblem and take a corresponding step at each iteration. The trust-region BFGS method we discuss for illustration is a simple combination of the trust-region method in [11] with the BFGS algorithm in [9]. Our purpose is to understand the fundamental difference between these two different strategies in nonsmooth optimization.

We organize this paper as follows. Section 3 shows how the exact-line-search BFGS method succeeds on a representative convex nonsmooth function. We also provide numerical evidence for linear convergence of an inexact-line-search BFGS method on the same example. Section 4 presents an illustrative proof that the inexact-line-search BFGS method cannot stall at a spurious limit point, when applied to a representative nonsmooth function without any stationary points, in contrast with the method of steepest descent. Section 5 gives an example where the inexact-line-search BFGS method converges to a point with descent directions. (This example does not disprove the challenge question from [9], since the limit point is nonetheless Clarke stationary.) In Sect. 6, we discuss possible reasons why the line-search BFGS method seems so much more successful than the trust-region method when applied to nonsmooth functions.

2 Line-Search BFGS Method

In this paper, we study the BFGS and line search algorithms described in [9]. Unless otherwise stated, we consider vectors in this paper as column vectors, and we denote the transpose of a column vector x by x^T. The line-search BFGS method applied to minimize a function $f : \mathbb{R}^n \to \mathbb{R}$ iterates as follows. We use x_k, H_k, and p_k to denote the current point, the approximate inverse Hessian matrix, and the line search direction at the kth iteration. We begin with an initial point x_0 and an initial positive semidefinite matrix H_0 and, then, repeatedly execute the following loop. We stop if we encounter an iterate x_k where the gradient $\nabla f(x_k)$ is either zero or does not exist.

Algorithm (Line-search BFGS)

repeat

Search direction: $p_k := -H_k \nabla f(x_k)$;

Step length: $x_{k+1} := x_k + \alpha_k p_k$, where α_k satisfies the following conditions,

for fixed $c_1 < c_2$ in $(0, 1)$:

$$f(x_k + \alpha_k p_k) \le f(x_k) + c_1 \alpha_k \nabla f(x_k)^T p_k \tag{1}$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \ge c_2 \nabla f(x_k)^T p_k; \tag{2}$$

Gradient increment: $y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$;

Inverse Hessian factor: $V_k := I - (p_k^T y_k)^{-1} p_k y_k^T$;

Inverse Hessian update: $H_{k+1} := V_k H_k V_k^T + \alpha_k (p_k^T y_k)^{-1} p_k p_k^T$;

Iteration count: $k = k + 1$;

end(repeat)

In the line search, condition (1) is usually called the *Armijo condition*, while Condition (2) is called the *weak Wolfe condition*. This weak Wolfe condition is the appropriate choice in the nonsmooth case, rather than the "strong" Wolfe condition: for more discussion, see [9].

Well-known properties of the BFGS method include the *secant condition*

$$s_k := x_{k+1} - x_k = H_{k+1} y_k,$$

and the fact that the matrix H_k remains positive definite. For simplicity, we use the abbreviated notation $\nabla f_k := \nabla f(x_k)$.

3 BFGS with Exact Line Search

In this section, we will give a full analysis of the BFGS method with an exact line search (instead of the weak Wolfe version above), applied to one particular representative nonsmooth example. The exact line search step length is chosen by $\alpha_k \in \mathrm{argmin}_\alpha \{f(x_k + \alpha p_k)\}$. Note that the BFGS method we consider here stops whenever it encounters a nonsmooth point (by which we mean a point at which f is not differentiable).

We begin with a structural property of the exact-line-search BFGS method. For simplicity, we state the result for infinite sequences of iterates.

Proposition 3.1 (Exact-line-search BFGS sequences) *Consider a function f on $\mathbb{R}^n$ and a sequence of points $x_0, x_1, x_2 \ldots$ in $\mathbb{R}^n$ at which f is differentiable and noncritical. Define vectors $s_k := x_{k+1} - x_k$ and $y_k := \nabla f_{k+1} - \nabla f_k$, for $k = 0, 1, 2, \ldots$. If (x_k) is an exact-line-search BFGS sequence for f, then the following properties hold for all $k = 0, 1, 2, \ldots$:*

1. $\nabla f_{k+1}^T s_k = 0$
2. $y_k^T s_{k+1} = 0$
3. $\nabla f_k^T s_k < 0.$

Conversely, suppose $n = 2$, and f is convex. If $\nabla f_0^T s_0 < 0$, and Properties 1 and 2 hold, along with the property $s_k \neq 0$, for all $k = 0, 1, 2, \ldots$, then (x_k) is an exact-line-search BFGS sequence for f.

Proof The forward direction is well known and routine, as follows. Property 1 follows from the definition of the exact line search. To see Property 2, note

$$y_k^T s_{k+1} = \alpha_{k+1} y_k^T p_{k+1} = -\alpha_{k+1} y_k^T H_{k+1} \nabla f_{k+1} = -\alpha_{k+1} s_k^T \nabla f_{k+1} = 0,$$

using the secant condition and Property 1. Property 3 follows from the fact that H_k is positive definite.

We prove the converse by induction. Since $\nabla f(x_0)^T s_0 < 0$, we can find a positive definite matrix H_0 such that $s_0 = -H_0 \nabla f_0$. The BFGS method then defines $p_0 = s_0$, and the exact line search then seeks α_0 minimizing $f(x_0 + \alpha p_0)$. Since f is convex and $\nabla f_1^T p_0 = 0$, it follows that $\alpha_0 = 1$ is a minimizer, and, hence, x_1 is a valid choice for the next BFGS iterate.

Now suppose we have proved that the iterates $x_0, x_1, x_2, \ldots, x_k$ constitute a valid BFGS sequence. Then, using the definition of the search direction p_k, the secant condition, and Property 1, we know that

$$p_k^T y_{k-1} = -\nabla f_k^T H_k y_{k-1} = -\nabla f_k^T s_{k-1} = 0.$$

On the other hand, Property 2 shows $s_k^T y_{k-1} = 0$. By assumption, the vectors s_{k-1}, s_k, and ∇f_k are all nonzero; by the secant property, so is y_{k-1}, and since H_k is positive definite, so is p_k. Since we are working on the space $\mathbb{R}^2$, and both the vectors p_k and s_k are orthogonal to y_{k-1}, they must be collinear. Then, the conditions that f is convex and $\nabla f_{k+1}^T s_k = 0$ imply that the point x_{k+1} minimizes the function f along the line $x_k + \mathbb{R} p_k$. Hence, x_{k+1} is a valid next iterate for the method. Therefore, the claim follows. □

The converse direction conveniently checks iterates of the exact-line-search BFGS on $\mathbb{R}^2$ without keeping track of the matrix H_k, as we see next.

3.1 A Parametrized Example

We next consider a simple but illustrative example on $\mathbb{R}^2$: for fixed $a > 0$,

$$f([u\ v]^T) = u^2 + \max\{v, -av\}. \tag{3}$$

This function has a global minimizer at zero and is nonsmooth at every point on one axis (and is "partly smooth" [12] relative to that axis). If we initialize appropriately, the algorithm generates points alternating between two parabolas and converging linearly to the optimal solution. In all other cases, the algorithm will terminate at a nonsmooth point after finitely many iterations.

Proposition 3.2 *Consider the exact-line-search BFGS method, applied to minimize the function (3), initialized with*

$$[u_0\ v_0] = \left[1\ \ \frac{2}{a^2+3a+1}\right] \quad \text{and} \quad H_0 = \begin{bmatrix} \frac{a}{2(a+1)} & 0 \\ 0 & \frac{2}{(a+1)^2} \end{bmatrix}.$$

The iterates converge linearly to the unique global minimizer zero, with rate $\rho = \frac{a}{(a+1)^2}$*, and oscillate between the two parabolas*

$$v = \frac{2}{a^2+3a+1}u^2 \quad \text{and} \quad v = -\frac{2a}{a^2+3a+1}u^2. \tag{4}$$

Explicitly, the iterates are given by

$$[u_{2k}\ v_{2k}] = \left[\rho^k\ \ \frac{2\rho^{2k}}{a^2+3a+1}\right], \quad [u_{2k+1}\ v_{2k+1}] = \left[\frac{\rho^k}{a+1}\ \ -\frac{2\rho^{2k+1}}{a^2+3a+1}\right].$$

Moreover, the corresponding inverse Hessian approximations are, for $k > 0$,

$$H_1 = \frac{1}{2(a^2+a+1)} \begin{bmatrix} 2a^2+a & 2a(1-a)(1+a)^{-1} \\ 2a(1-a)(1+a)^{-1} & 4(a^3+a+1)(1+a)^{-3} \end{bmatrix}$$

$$H_{2k} = \frac{1}{2a^2(a^2+a+1)} \begin{bmatrix} a^2(a^2+2a+2) & 2a\rho^k \\ 2a\rho^k & 4(a^2+1)\rho^{2k} \end{bmatrix}$$

$$H_{2k+1} = \frac{1}{2(1+a)^2(a^2+a+1)} \begin{bmatrix} (1+a)^2(2a^2+2a+1) & -2a^2(1+a)\rho^k \\ -2a^2(1+a)\rho^k & 4(a^2+1)\rho^{2k} \end{bmatrix}.$$

The step sizes, for $k > 0$, *are* $\alpha_0 = 1$, $\alpha_1 = \frac{1}{a(1+a)}$, $\alpha_{2k} = a\rho$, *and* $\alpha_{2k+1} = \frac{\rho}{a}$.

Proof A calculation verifies that the given sequence of iterates is an exact-line-search BFGS sequence, by Proposition 3.1. Since the function is strictly convex along each search direction, the given sequence is the unique exact-line-search BFGS sequence under the given initialization. The formulae for the inverse Hessian approximations are easy to verify directly by induction. □

Note that the convergence rate ρ is unchanged under the transformation $a \leftarrow \frac{1}{a}$. This is not surprising, given the invariance of the method under scaling of the objective, and a consequent simple symmetry property.

In the example above, for very specific initial values, BFGS generates a sequence of points oscillating between two parabolas and converging linearly to the optimal solution, zero. We also observe, at each iteration, that the method crosses the axis on which the function is nonsmooth. Seemingly this property allows BFGS to "learn" the nonsmooth structure of the problem, coded into the inverse Hessian approximations. By contrast, as we see next, under general initial conditions for this function, except in essentially the case above, the exact line search causes the simple nonsmooth BFGS method we consider here to break down at a nonsmooth point that is not optimal.

Proposition 3.3 *Consider the exact-line-search BFGS method applied to minimize the function (3). Unless the first two iterates* $[u_0\ v_0]^T$ *and* $[u_1\ v_1]^T$ *satisfy* $u_1 = (1+a)^{-1}u_0$ *and* $[u_1\ v_1]^T$ *lies on one of the two parabolas described by Eq. (4), the algorithm eventually stops at a nonsmooth point.*

Proof For simplicity, we focus on the case $a = 1$. Assume the method generates an infinite sequence of points $x_k = [u_k\ v_k]^T$ for $k = 1, 2, 3, \ldots$ at which f is smooth. We first claim that the coordinate v_k must change sign at every iteration. If not, then, without loss of generality, there is an iteration n with $v_{n-1} < 0$ and $v_n < 0$. The previous result ensures $(\nabla f_n - \nabla f_{n-1})^T(x_{n+1} - x_n) = 0$, so the search direction p_n must be in the direction of the vector $[0\ 1]^T$. But the exact line search then causes termination at the nonsmooth point $x_{n+1} = [u_n\ 0]^T$, contradicting our assumption.

Without loss of generality, we can next assume $v_{2k} > 0$, $v_{2k+1} < 0$ for all $k = 1, 2, 3, \ldots$. By applying the previous result, we easily arrive at the recursion

$$u_{2k+1} = -\frac{u_{2k} - u_{2k-1}}{2}, \quad v_{2k+1} = v_{2k} + \frac{(u_{2k} - u_{2k-1})(3u_{2k} - u_{2k-1})}{2},$$

and similarly

$$u_{2k} = -\frac{u_{2k-1} - u_{2k-2}}{2}, \quad v_{2k} = v_{2k-1} + \frac{(u_{2k-1} - u_{2k-2})(-3u_{2k-1} + u_{2k-2})}{2}.$$

Hence, we deduce $u_n = -(u_{n-1} - u_{n-2})/2$ for all iterates $n > 2$, and consequently $u_n + u_{n-1} = (u_{n-1} + u_{n-2})/2$. By induction, we have $u_n + u_{n-1} = 2^{1-n}(u_1 + u_0)$. We deduce, for $k = 1, 2, 3, \ldots$,

$$u_{2k} = \frac{2}{3}\Big(\frac{1}{2^{2k}} - 1\Big)(u_1 + u_0) + u_0, \quad u_{2k+1} = \frac{2}{3}\Big(\frac{1}{2^{2k+1}} + 1\Big)(u_1 + u_0) - u_0.$$

In particular, $u_{2k} \to \lambda := \frac{1}{3}u_0 - \frac{2}{3}u_1$ and $u_{2k+1} \to -\lambda$ as $k \to \infty$.

Now assume $\frac{u_1}{u_0} \neq \frac{1}{2}$, so $\lambda \neq 0$. Suppose first that $\lambda > 0$. (The case $\lambda < 0$ is similar.) Then, for all large k we must have $u_{2k-1} < 0$ and $u_{2k} > 0$. By the previous result we know $(\nabla f_{2k} - \nabla f_{2k-1})^T(x_{2k+1} - x_{2k}) = 0$, so the search direction p_{2k} is in the direction $[-1\ \mu]^T$ where $\mu = u_{2k} - u_{2k-1}$. By definition of the exact line search, we know $x_{2k+1} = x_{2k} + \beta[-1\ \mu]^T$, where the scalar β minimizes $(u_{2k} - \beta)^2 + |v_{2k} + \beta\mu|$.

If $v_{2k} + \beta\mu \geq 0$, then either $v_{2k+1} = 0$ or $v_{2k+1} > 0$. In the case $v_{2k+1} = 0$, our method stops at this nonsmooth point. If, on the other hand, $v_{2k+1} > 0$, then the same argument shows $v_{2k+2} = 0$.

Suppose, on the other hand, $v_{2k} + \beta\mu < 0$. Then, by its definition, β minimizes $(u_{2k} - \beta)^2 - (v_{2k} + \beta\mu)$, so a calculation shows $\beta = (3u_{2k} - u_{2k-1})/4$.

Since $u_{2k} > 0$ and $u_{2k-1} < 0$, we deduce $\beta > 0$. We also know $v_{2k} > 0$ and $\mu > 0$, so $v_{2k+1} = v_{2k} + \beta\mu > 0$, which contradicts the property $v_{2k+1} < 0$.

Now consider the final case, where $\frac{u_1}{u_0} = \frac{1}{2}$ but $v_1 \neq -\frac{2}{5}u_1^2$. Then, the formula above for the component u_n reduces to $u_n = 2^{-n}u_0$, and we can similarly deduce a formula for the component v_n:

$$v_{2k+1} = v_{2k} + \frac{(u_{2k} - u_{2k-1})(3u_{2k} - u_{2k-1})}{2} = v_{2k} - \frac{u_0^2}{2^{4k+1}}$$

$$v_{2k} = v_{2k-1} - \frac{(u_{2k-1} - u_{2k-2})(3u_{2k-1} - u_{2k-2})}{2} = v_{2k-1} + \frac{u_0^2}{2^{4k-1}}.$$

Hence, v_n converges to $v_1 + \frac{2}{5}u_1^2$, which, by assumption, is nonzero. Thus, v_n eventually does not change sign, which quickly gives a contradiction. □

Corollary 3.1 *Consider the exact-line-search BFGS method applied to minimize the function (3). If the method does not terminate, then iterates must oscillate between the two parabolas described by Eq. (4) and converge linearly to the global minimizer zero.*

Proof Again we concentrate on the case $a = 1$ for simplicity. By the previous result, we can assume $v_{2k} > 0$, $v_{2k+1} < 0$, $u_0 = 2u_1$, and $v_1 = -\frac{2}{5}u_1^2$. Then, we deduce the formulae $u_{2k} = 2^{-2k}u_0$, $v_{2k} = \frac{2}{5}4^{-2k}u_0^2$, $u_{2k+1} = 2^{-(2k+1)}u_0$, and $v_{2k+1} = -\frac{2}{5}4^{-(2k+1)}u_0^2$, so the claim follows. □

To summarize this very simple theoretical case study, we observe two possible cases. Either the exact-line-search BFGS method converges linearly to the global minimizer zero, oscillating between two parabolas, or the line search causes the method to terminate prematurely at a nonoptimal nonsmooth point.

4 BFGS with Inexact Line Search

We turn next from the idealized version of BFGS of the previous section to a more realistic version with an inexact line search. Again we focus on very simple examples, seeking insight on the method in the nonsmooth case, rather than extensive practical experience. For the latter, see Lewis and Overton [9], and the references therein.

Recall that, for minimizing the function f, with the current iterate x_k and search direction p_k, the line search seeks a step length α_k: that is, a scalar t satisfying the Armijo and Wolfe conditions, namely

$$f(x_k + tp_k) \leq f(x_k) + c_1 t \nabla f(x_k)^T p_k \tag{5}$$

$$\nabla f(x_k + tp_k)^T p_k \geq c_2 \nabla f(x_k)^T p_k. \tag{6}$$

We here use the following algorithm [9, Alg. 2.6] to find a step length.

Algorithm (Inexact line search)

$\alpha_{\min} \leftarrow 0$;

$\alpha_{\max} \leftarrow +\infty$;

$\alpha \leftarrow 1$;

repeat

  if inequality (5) fails, $\alpha_{\max} \leftarrow \alpha$

  else if inequality (6) fails, $\alpha_{\min} \leftarrow \alpha$

  else stop;

  if $\alpha_{\max} < +\infty$, $\alpha \leftarrow (\alpha_{\min} + \alpha_{\max})/2$

  else $\alpha \leftarrow 2\alpha_{\min}$;

end(repeat)

As we saw in the previous section, for nonsmooth examples, the behavior of BFGS with an exact line search can depend on the initialization. By contrast, the behavior with an inexact line search in practice seems more robust. We consider here the following question: to what extent we can gain insight on the convergence rate (*with respect to the number of function evaluations carried out in the inexact line search*) from the behavior with an exact line search?

4.1 The Parametrized Example: Ill Conditioning

For illustration, we return to the previous example, the function (3). We are particularly interested in the broad dependence of the rate of convergence on the parameter a, which gives a certain measure of the "conditioning" of the problem. Notwithstanding the dependence of the standard smooth theory on the assumption $c_1 > 0$ for simplicity of exposition, we take $c_1 = 0$ and $c_2 = 0.9$ here.[1] Numerical experimentation shows that, with random initialization, the inexact-line-search BFGS eventually crosses the line $v = 0$ at each iteration, and has a linear convergence rate, plotted in red on Fig. 1. (This behavior is relatively insensitive to the choice of c_2.)

A reasonable fit to the observed linear convergence rate is given by the function $r(a)$, defined for $0 < a < 1$ by

$$\log_2\big(r(a)\big) = \frac{\log_2(3a^2+3a+1) - \log_2((a^2+3a+3)(a+1)^2)}{2\log_2(1+\frac{1}{a})}, \tag{7}$$

and for $a > 1$ by the symmetry $r(a) = r(1/a)$. This function is plotted in blue on Fig. 1. We arrive at this rough fit through the following loose intuition.

As Proposition 3.2 indicates, when applying the exact-line-search BFGS to this function with appropriate starting points, we generate the iterates

$$x_{2k} = \left[\rho^k \quad \frac{2\rho^{2k}}{a^2+3a+1}\right]^T \quad \text{and} \quad x_{2k+1} = \left[\frac{\rho^k}{a+1} \quad -\frac{2\rho^{2k+1}}{a^2+3a+1}\right]^T \tag{8}$$

(where $\rho = \frac{a}{(a+1)^2}$), with step lengths $\alpha_{2k} = \left(1+\frac{1}{a}\right)^{-2}$ and $\alpha_{2k+1} = (1+a)^{-2}$.

The linear convergence rate per iteration is, in this case,

$$\frac{f(x_{2k+1})}{f(x_{2k})} = \frac{1+\frac{2a^2}{a^2+3a+1}}{1+\frac{2}{a^2+3a+1}}\frac{1}{(a+1)^2} \quad \text{and} \quad \frac{f(x_{2k+2})}{f(x_{2k+1})} = \frac{1+\frac{2}{a^2+3a+1}}{1+\frac{2a^2}{a^2+3a+1}}\frac{a^2}{(1+a)^2},$$

for the even and odd iterations, respectively.

Since we set $c_1 = 0$, the exact line search step length in Sect. 3 also satisfies the line search conditions. Consider the case when $a > 0$ is small. In that case, the odd iterations generate a large decrease in function value with a step length close to one. By contrast, the even iterations generate only a small decrease, and, to do so, need to use a small step length $(1+1/a)^{-2}$. We might expect a bisection-based line search to need roughly $\log_2\big((1+1/a)^2\big)$ function evaluations to locate the step.

Consider the ratio r_{exact} by which the function value decreases during the course of an iteration. Let us denote by q the number of function evaluations (or "trials") during the inexact line search and by r the average ratio of decrease per function evaluation. Then, we have the relationship $r^q = r_{\text{exact}} < 1$. Hence, an estimate of the convergence rate during those iterations is

[1] See the discussion of this point in [9, pp. 151–154].

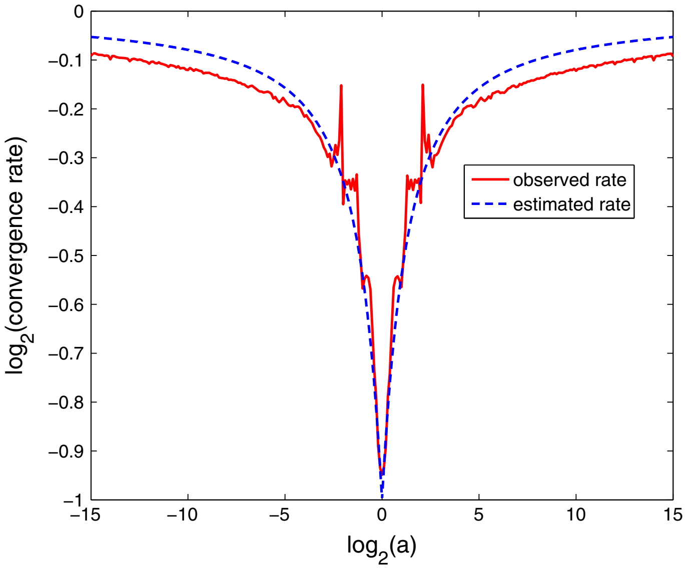


Fig. 1 A fit of convergence rate

$$r(a) = \left(\frac{f(x_{2k+1})}{f(x_{2k})}\right)^{\theta}, \quad \text{where} \quad \theta = \frac{1}{2\log_2(1+\frac{1}{a})}. \tag{9}$$

Experiments with the inexact-line-search BFGS suggest that iterations analogous to the one above are typical throughout the run and not just for alternating values of k. Hence, we arrive at the estimate (7), which does give a reasonable fit to the experimental data. A similar argument applies to large $a > 0$.

We can explore this behavior in a more controlled fashion. Consider, for the moment, the behavior of our inexact line search when started at the exact-line-search iterates x_{2k} (or x_{2k+1}) described by (8) and searching in the corresponding directions p_{2k} (or p_{2k+1}). Numerical results (with $c_0 = 0, c_1 = 0.9$) suggest that the number of function evaluations needed by the line search depends only on a and does not depend on the iteration count k. The following subsidiary result throws some light on that dependence.

Proposition 4.1 *Consider the function (3), with parameter $a = 2^m - 1$ (for $m = 1, 2, 3, \ldots$), and the exact-line-search BFGS iterates (8) with search directions p_{2k} and p_{2k+1}. With those iterates and search directions, the inexact line search would generate the step lengths $\alpha_{2k} = 1$ and $\alpha_{2k+1} = 2^{-m}$. On the other hand, in the case $a = 1/(2^m - 1)$, we obtain $\alpha_{2k} = 2^{-m}$ and $\alpha_{2k+1} = 1$.*

Proof We only prove the case when $a > 1$. The proof for $a < 1$ is similar. For the even iterations, since the iterate is $x_{2k} = \begin{bmatrix} \rho^k & \frac{2\rho^{2k}}{a^2+3a+1} \end{bmatrix}^T$, and the direction is $p_{2k} = \begin{bmatrix} -\frac{(a+1)\rho^k}{a} & -\frac{2\rho^{2k}}{a^2} \end{bmatrix}^T$, we deduce

$$f(x_{2k}) = \Big(1 + \frac{2}{a^2+3a+1}\Big)\rho^{2k}, \quad f(x_{2k}+p_{2k}) = \Big(\frac{3}{a^2} - \frac{2}{a^2+3a+1}\Big)\rho^{2k},$$
$$\nabla f(x_{2k})^T p_{2k} = -\Big(2 + \frac{2}{a} + \frac{2}{a^2}\Big)\rho^{2k}, \quad \nabla f(x_{2k}+p_{2k})^T p_{2k} > 0.$$

Since $\alpha = 1$ satisfies the line search conditions, we deduce $\alpha_{2k} = 1$. For the odd iterations, we have $x_{2k+1} = \begin{bmatrix} \frac{\rho^k}{a+1} & -\frac{2\rho^{2k+1}}{a^2+3a+1}\end{bmatrix}^T$ and $p_{2k+1} = \begin{bmatrix}-\rho^k & 2\rho^{2k+1}\end{bmatrix}^T$. Then, we obtain

$$\begin{aligned}
f(x_{2k+1}) &= \Big(\frac{1}{(a+1)^2} + \frac{2a\rho}{a^2+3a+1}\Big)\rho^{2k},\\
x_{2k+1} + \alpha p_{2k+1} &= \Big[\Big(\frac{1}{a+1} - \alpha\Big)\rho^k \quad \Big(-\frac{2}{a^2+3a+1} + 2\alpha\Big)\rho^{2k+1}\Big]^T,\\
\nabla f(x_{2k+1})^T p_{2k+1} &= -\Big(\frac{2}{a+1} + 2a\rho\Big)\rho^{2k}.
\end{aligned}$$

Consider the case $\alpha = 2^{-l}$ for some integer $l = 1, 2, 3, \ldots, m-1$. We have

$$\nabla f(x_{2k+1} + \alpha p_{2k+1})^T p_{2k+1} \;=\; -\Big(\frac{2}{a+1} - 2\alpha\Big)\rho^{2k} + 2\rho^{2k+1} \;>\; 0.$$

Note that $\alpha \geq \frac{2}{a+1}$, so

$$\begin{aligned}
f(x_{2k+1} + \alpha p_{2k+1}) &= \Big(\alpha - \frac{1}{a+1}\Big)^2\rho^{2k} + \Big(2\alpha - \frac{2}{a^2+3a+1}\Big)\rho^{2k+1}\\
&\geq \Big(\frac{2}{a+1} - \frac{1}{a+1}\Big)^2\rho^{2k} + \Big(\frac{2}{a+1} - \frac{2}{a^2+3a+1}\Big)\rho^{2k+1}\\
&> \frac{1}{(a+1)^2}\rho^{2k} + \frac{2a}{a^2+3a+1}\rho^{2k+1} \;=\; f(x_{2k+1}).
\end{aligned}$$

Therefore, the line search algorithm will successively try $\alpha = 1, \frac{1}{2}, \ldots, \frac{1}{2^{m-1}}$, and finally $\alpha = \frac{1}{2^m} = \frac{1}{a+1}$. At this point we have

$$\begin{aligned}
\nabla f(x_{2k+1} + \alpha p_{2k+1})^T p_{2k+1} &= -\Big(\frac{2}{a+1} - 2\alpha\Big)\rho^{2k} + 2\rho^{2k+1} > 0,\\
f(x_{2k+1} + \alpha p_{2k+1}) &= \Big(\alpha - \frac{1}{a+1}\Big)^2\rho^{2k} + \Big(2\alpha - \frac{2}{a^2+3a+1}\Big)\rho^{2k+1}\\
&\leq \frac{2a}{(a+1)^2}\rho^{2k+1} \;\leq\; f(x_{2k+1}),
\end{aligned}$$

so the line search conditions are satisfied. The claim follows. □

The result above suggests that, if we were following the iterates generated by the exact-line-search BFGS, then, for small $a > 0$, the "work" involved in each iteration, measured loosely by the number of function evaluations our *inexact* line search would take, is dominated by the even iterations, for which it depends on the factor $\log_2(1+\frac{1}{a})$, a key ingredient of the estimate (7).

4.2 Example: A Ridge

Rigorous general analysis of the inexact-line-search BFGS in the nonsmooth case seems challenging. Here, for reassurance, we prove one very modest result. In the simplest possible case, a maximum of two affine functions (a "ridge"), we can at least be sure that the method will not converge to a spurious limit.

By way of contrast, the method of steepest descent (where the search direction p_k is just $-\nabla f_k$), with our inexact line search, can easily fail on such functions. For example, for the function $f([x_1\ x_2]^T) = 6|x_1| + 3x_2$, the steepest descent using the same inexact line search (with $0 \le c_1 \le \frac{1}{3}$) and starting at the initial point $[2\ 3]^T$, generates the iterates $2^{-k}[2(-1)^k\ 3]^T$, $k = 1, 2, \ldots$. These iterates converge to the origin which is not a critical point for the convex function f. Furthermore, the convergence is sublinear in the number of function evaluations: at iteration k, the line search requires k bisections. In contrast, BFGS with the same line search rapidly reduces the function value toward $-\infty$. On such examples, the following result captures the general argument.

Proposition 4.2 *If the inexact-line-search BFGS method applied to the function $f([u\ v]^T) = |u| + v$ generates a sequence of iterates $x_k = [u_k\ v_k]^T$ with $u_k \neq 0$ (for $k = 0, 1, 2, \ldots$), then x_k does not converge.*

Proof The line search guarantees that, if the current point satisfies $u_k > 0$, then the search direction $p_k = [m_k\ l_k]^T$ satisfies $m_k < 0$, and at the next iteration we must have $u_{k+1} < 0$. A similar argument holds if $u_k < 0$.

We first prove $|m_k| > |l_k|$ for all iterations k. Without loss of generality, suppose $u_k > 0$. Since $\nabla f(x_k) = [1\ 1]^T$, then $m_k + l_k = -[1\ 1]H_k[1\ 1]^T$. Note that $y_k = \nabla f(x_{k+1}) - \nabla f(x_k) = [-2\ 0]^T$ and $V_k = I - (p_k^T y_k)^{-1} p_k y_k^T$ is the matrix $\begin{bmatrix} 0 & 0 \\ -l_k/m_k & 1 \end{bmatrix}$. Hence, using the notation $H_k = \begin{bmatrix} a_k & b_k \\ b_k & c_k \end{bmatrix}$, we see that the matrix $H_{k+1} = V_k H_k V_k^T + \alpha_k (p_k^T y_k)^{-1} p_k p_k^T$ is given by

$$\begin{bmatrix} -\frac{\alpha_k m_k}{2} & -\frac{\alpha_k l_k}{2} \\ -\frac{\alpha_k l_k}{2} & a_k(\frac{l_k}{m_k})^2 - 2b_k\frac{l_k}{m_k} + c_k - \alpha_k\frac{(l_k)^2}{2m_k} \end{bmatrix}.$$

Thus, $m_{k+1} = \frac{\alpha_k(l_k - m_k)}{2} > 0$, which implies $l_k - m_k > 0$. Combined with the fact that $m_k + l_k < 0$, we have $|m_k| > |l_k|$.

We now prove the proposition by contradiction. Suppose the sequence x_k converges. Then, $\alpha_k m_k \to 0$ and $\alpha_k l_k \to 0$. Note

$$p_{k+1} = -H_{k+1}\nabla f(x_{k+1}) = -\begin{bmatrix} -\frac{\alpha_k m_k}{2} & -\frac{\alpha_k l_k}{2} \\ -\frac{\alpha_k l_k}{2} & c_{k+1} \end{bmatrix}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\alpha_k(l_k - m_k)}{2} \\ -\frac{\alpha_k l_k}{2} - c_{k+1} \end{bmatrix},$$

where $c_{k+1} = a_k(\frac{l_k}{m_k})^2 - 2b_k\frac{l_k}{m_k} + c_k - \alpha_k\frac{(l_k)^2}{2m_k}$. Then, we deduce that $m_{k+1} = \frac{\alpha_k(l_k - m_k)}{2} \to 0$. We now show that the positive number c_{k+1} stays bounded away from zero.

To this end, note by induction that we have

$$H_{k+1} = V_k \dots V_0 H_0 V_0^T \dots V_k^T + \alpha_0 V_k \dots V_1 (p_0^T y_0)^{-1} p_0 p_0^T V_1^T \dots V_k^T \\ + \dots + \alpha_k (p_k^T y_k)^{-1} p_k p_k^T.$$

Since

$$V_i V_j = \begin{bmatrix} 0 & 0 \\ -\frac{l_i}{m_i} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -\frac{l_j}{m_j} & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -\frac{l_j}{m_j} & 1 \end{bmatrix} = V_j,$$

we obtain $H_{k+1} = V_0 H_0 V_0^T + \alpha_0 V_1 (p_0^T y_0)^{-1} p_0 p_0^T V_1^T + \dots + \alpha_k (p_k^T y_k)^{-1} p_k p_k^T$. Since $p_k^T y_k > 0$ for all k, the $(2, 2)$-entry of the matrix H_k is increasing in k, and, hence, is at least as large as that of the matrix $V_0 H_0 V_0^T$, namely the quantity $a_0\big(\frac{l_0}{m_0}\big)^2 - 2b_0\frac{l_0}{m_0} + c_0 > 0$, as required.

Finally, observe $l_{k+1} = -\frac{\alpha_k l_k}{2} - c_{k+1}$ cannot converge to zero. Since we know $|m_k| > |l_k|$, we obtain a contradiction. The claim follows. □

The idea of this proof extends to the maximum of any two affine functions on $\mathbb{R}^n$. Note too how this example illustrates behavior that seems to drive the success of BFGS in the nonsmooth case: the inexact line search crosses the line $u = 0$ (the manifold with respect to which the function is partly smooth) at each iteration, allowing the method to "learn" the nonsmooth structure.

5 A Limit Point with Descent Directions

In the above sections we illustrated good behavior of BFGS on some nonsmooth functions. We now contrast with an illustration of possible pitfalls.

The reference [9] poses the challenge: does the inexact-line-search BFGS method converge only to points that are *Clarke stationary*? For locally Lipschitz functions, this amounts to saying that we can find convex combinations of gradients at nearby points that are arbitrarily small. For a large class of functions (for example, those of the form $h\big(c(\cdot)\big)$ with h finite and convex and c smooth), Clarke stationarity guarantees that there exist no directions of linear descent. However, in general Clarke stationarity does not rule out descent directions: the function $x \mapsto -|x|$ at $x = 0$ is a simple example.

Here we show how BFGS can converge to a point at which there exist directions of linear descent. Suggestions of this phenomenon were recently observed numerically on an example of Nesterov by Gürbüzbalaban and Overton [13]. We begin with some relevant definitions (see [14]).

Definition 5.1 Consider a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and a point $\bar{x}$ with $f(\bar{x})$ finite. Consider a vector $v \in \mathbb{R}^n$.

1. We call v a *regular subgradient* of f at $\bar{x}$, written $v \in \hat{\partial} f(\bar{x})$, iff

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x} \rangle + o(x - \bar{x}) \quad \text{and} \quad x \to \bar{x};$$

2. We call $\bar{x}$ *regular stationary* iff $0 \in \hat{\partial} f(\bar{x})$. (In other words, $\bar{x}$ is a local minimizer, up to first order.)
3. We call v a *limiting subgradient* of f at $\bar{x}$, written $v \in \partial f(\bar{x})$, iff there are sequences $x^\nu \to \bar{x}$ with $f(x^\nu) \to f(\bar{x})$ and $v^\nu \in \partial f(x^\nu)$ with $v^\nu \to v$.
4. We call $\bar{x}$ *limiting stationary* iff $0 \in \partial f(\bar{x})$.
5. When f is Lipschitz around $\bar{x}$, and zero is a convex combination of limiting subgradients there, we call $\bar{x}$ *Clarke stationary*.

A direction $p \in \mathbb{R}^n$ satisfying $\limsup_{t \downarrow 0} \frac{1}{t}\big(f(\bar{x}+tp) - f(\bar{x})\big) < 0$ is called a *direction of linear descent*. (In this case, $\bar{x}$ is clearly not regular stationary.)

Reference [9, Proposition 3.3] gives an example of the exact-line-search BFGS applied to $f(x) = ||x||$ in $\mathbb{R}^2$. The complete statement is as follows.

Proposition 5.1 *Consider the exact-line-search BFGS method applied to the Euclidean norm in $\mathbb{R}^2$, initialized by*

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

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.

In fact, the use of our inexact line search (Algorithm 3) instead of the exact line search, generates the same points, as the following calculation shows.

Proposition 5.2 *Consider the inexact-line-search BFGS method applied to the Euclidean norm in $\mathbb{R}^2$. For any $0 < c_1 < \frac{2}{3}$ and $c_1 < c_2 < 1$, the method, initialized as in Proposition 5.1, generates a sequence of vectors x_k that rotate counterclockwise through an angle of $\frac{\pi}{3}$ and shrink by a factor $\frac{1}{2}$ at each iteration. For*

$$R = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix},$$

at the kth iteration we have:

$$x_k = 2^{-k} R^{-k} x_0, \quad \alpha_k = \frac{1}{4}, \quad p_k = 2^{-k} R^{-k} \begin{bmatrix} -3 \\ \sqrt{3} \end{bmatrix} \quad \text{and} \quad H_k = 2^{-k} R^{-k} H_0 R^k.$$

Proof A direct calculation (see [9, Theorem 3.2]) shows that the exact-line-search BFGS applied to the function $\|x\|$ initialized with x_0 and H_0 generates the sequence (x_k). It is also easy to check that the exact step size for each iteration is $\frac{1}{4}$. In order to prove the result, it is sufficient to prove that the step size for the inexact-line-search BFGS is also $\frac{1}{4}$ for each iteration.

Consider the kth iteration. The line search algorithm will try $t = 1$ first. Since $\nabla f(x_k)^T p_k = -3 \times 2^{-k}$ and $f(x_k + p_k) = \sqrt{7} \times 2^{-k}$, then

$$f(x_k + p_k) = 2^{-k}\sqrt{7} > 2^{-k}(1 - 3c_1) = f(x_k) + c_1 \nabla f(x_k)^T p_k$$
$$\nabla f(x_k + p_k)^T p_k = \frac{9}{\sqrt{7}} \times 2^{-k} \geq -3 \times 2^{-k} c_2 = -c_2 \nabla f(x_k)^T p_k.$$

Hence, the algorithm will try $t = \frac{1}{2}$. This time we note

$$f\left(x_k + \frac{1}{2}p_k\right) = 2^{-k} > 2^{-k}\left(1 - \frac{3}{2}c_1\right) = f(x_k) + \frac{c_1}{2}\nabla f(x_k)^T p_k$$
$$c_2 \nabla f(x_k)^T p_k = -3 \times 2^{-k} c_2 < 3 \times 2^{-k} = \nabla f(x_k + \frac{1}{2}p_k)^T p_k.$$

Now the algorithm will try $t = \frac{1}{4}$. We observe, since $c_1 < 2/3$,

$$f(x_k + \frac{1}{4}p_k) = 2^{-(k+1)} < 2^{-k}(1 - \frac{3}{4}c_1)\ (c_1 < \frac{2}{3}) = f(x_k) + \frac{c_1}{4}\nabla f(x_k)^T p_k$$
$$c_2 \nabla f(x_k)^T p_k = -3 \times 2^{-k} c_2 < 0 = \nabla f(x_k + \frac{1}{4}p_k)^T p_k.$$

We deduce $\alpha_k = 1/4$. The claim follows. □

The inexact-line-search BFGS thus only visits points on the half-lines given by $\mathbb{R}_+[\cos\frac{n\pi}{3}\ \ \sin\frac{n\pi}{3}]^T$ (for integers n). To construct an example where the algorithm converges to a point with descent directions, we ensure that BFGS still only visits those points, but change the function values elsewhere.

Proposition 5.3 *Consider the inexact-line-search BFGS applied to the function defined by* $g([u\ \ v]^T) = \sqrt{u^2 + v^2} \cdot \cos(18 \arctan \frac{v}{u})$ *(for* $[u\ \ v]^T \neq [0\ 0]^T$*) and* $g([0\ 0]^T) = 0$*, or equivalently in polar coordinates* $(r, \theta) \mapsto r\cos(18\theta)$*. For any* $0 < c_1 < \frac{2}{3}$ *and* $c_1 < c_2 < 1$*, if we initialize as in Proposition 5.1, then the method generates the same sequence as in Proposition 5.2 and, hence, converges to the point zero, at which there exist directions of linear descent.*

Proof The existence of directions of linear descent at zero is clear (for example, the ray $\theta = \pi/18$), so we simply need to prove that the BFGS method generates the same sequence (x_k) as in Proposition 5.2 by induction. Since

$$R = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \cos\frac{\pi}{3} & \sin\frac{\pi}{3} \\ -\sin\frac{\pi}{3} & \cos\frac{\pi}{3} \end{bmatrix} \quad \text{and} \quad R^{-1} = \begin{bmatrix} \cos\frac{\pi}{3} & -\sin\frac{\pi}{3} \\ \sin\frac{\pi}{3} & \cos\frac{\pi}{3} \end{bmatrix},$$

we deduce

$$R^k = \begin{bmatrix} \cos\frac{k\pi}{3} & \sin\frac{k\pi}{3} \\ -\sin\frac{k\pi}{3} & \cos\frac{k\pi}{3} \end{bmatrix} \quad \text{and} \quad R^{-k} = \begin{bmatrix} \cos\frac{k\pi}{3} & -\sin\frac{k\pi}{3} \\ \sin\frac{k\pi}{3} & \cos\frac{k\pi}{3} \end{bmatrix}.$$

Then, $g(x_k) = g(2^{-k}R^{-k}x_0) = 2^{-k} = f(x_k)$. Furthermore, we have

$$\left.\frac{\partial g(x)}{\partial u}\right|_{x=x_k} = \frac{u_k}{||x_k||}\cos\left(18\arctan\frac{v_k}{u_k}\right) = \frac{u_k}{||x_k||}$$

and

$$\left.\frac{\partial g(x)}{\partial v}\right|_{x=x_k} = \frac{v_k}{||x_k||}\cos(18\arctan\frac{v_k}{u_k}) = \frac{v_k}{||x_k||},$$

so $\nabla g(x_k) = \frac{x_k}{||x_k||} = \nabla f(x_k)$.

It suffices to show that, at each iteration k, the step size is always $\frac{1}{4}$. The idea of the proof is to compare the functions $f = \|\cdot\|$ and g along the search directions at each iteration and observe that the calculations during the inexact line search are identical. Figure 2 illustrates the idea, by plotting the functions f and g along a typical search direction.

Clearly $p_0 = -H_0\nabla g(x_0) = [-3\ \sqrt{3}]^T$ and $\nabla g(x_0) = \nabla f(x_0) = [1\ 0]^T$. The line search algorithm applied to g first tries $t = 1$. Since

$$\begin{aligned} g(x_0 + p_0) = g([-2\ \sqrt{3}]^T) &= \sqrt{7}\cos(18\text{arctan}\frac{-\sqrt{3}}{2}) \\ &> 1 = g(x_0) > g(x_0) + c_1\nabla g(x_0)^T p_0, \end{aligned}$$

we see $t = 1$ does not satisfy the Armijo condition. Moreover, as Fig. 2 illustrates, $\nabla g(x_0 + p_0)^T p_0 \geq \nabla f(x_0 + p_0)^T p_0$. Since, furthermore, $\nabla f(x_0 + p_0)^T p_0 \geq c_2\nabla f(x_0)^T p_0 = c_2\nabla g(x_0)^T p_0$, the value $t = 1$ satisfies the weak Wolfe condition for g. Therefore, the line search algorithm will try $t = \frac{1}{2}$.

As Fig. 2 indicates, we have $g(x_0 + \frac{1}{2}p_0) = f(x_0 + \frac{1}{2}p_0)$ and $\nabla g(x_0 + \frac{1}{2}p_0) = \nabla f(x_0 + \frac{1}{2}p_0)$. Hence, $t = \frac{1}{2}$ does not satisfy the Armijo condition but does satisfy the Wolfe condition, following Proposition 5.2. Hence, the line search will next try $t = \frac{1}{4}$. Since $g(x_0) = f(x_0)$, $\nabla g(x_0) = \nabla f(x_0)$, $g\big(x_0 + \frac{1}{4}p_0\big) = f\big(x_0 + \frac{1}{4}p_0\big)$, and $\nabla g\big(x_0 + \frac{1}{4}p_0\big) = \nabla f\big(x_0 + \frac{1}{4}p_0\big)$, it follows that $t = \frac{1}{4}$ satisfies the line search conditions. Hence, the iterates coincide for $k = 1$.

We now proceed inductively, in similar fashion. We suppose that up to kth iteration the iterates coincide, and, furthermore, $p_k = 2^{-k}R^{-k}p_0$ and $H_k = R^{-k}H_0R^k$. We want to prove coincidence at the $(k+1)$th iteration. First note that $x_k + tp_k = 2^{-k}R^{-k}(x_0 + tp_0)$ can be obtained by rotating $2^{-k}(x_0 + tp_0)$ counter-clockwise through an angle of $\frac{k\pi}{3}$. Then, we have $f(x_k + tp_k) = 2^{-k}f(x_0 + tp_0)$ and $g(x_k + tp_k) = 2^{-k}g(x_0 + t_0p_0)$. Therefore, by the above argument, the line search step

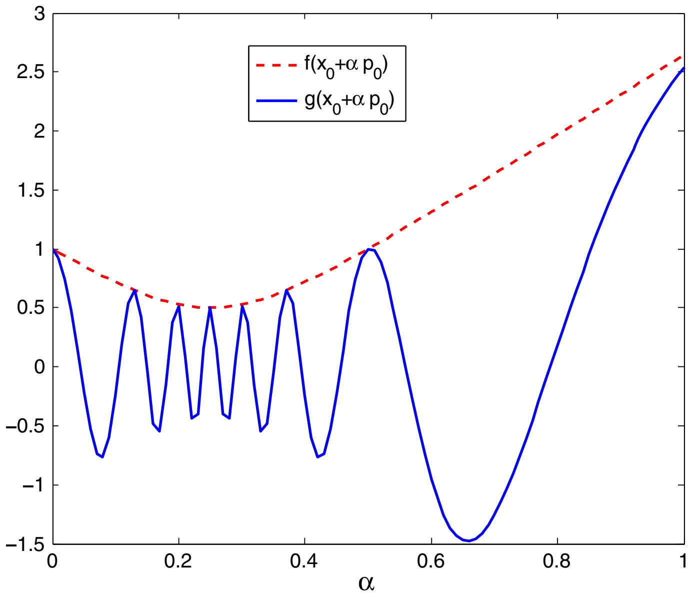


Fig. 2 A comparison of f and g along search direction p_0

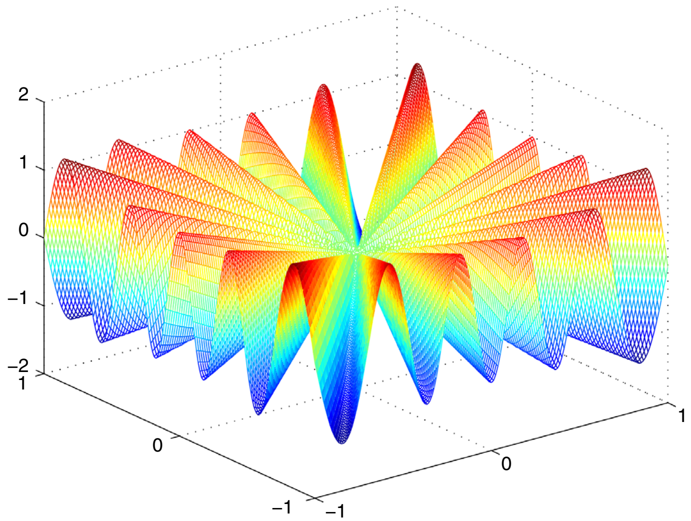


Fig. 3 Graph of the function g

size should be $\alpha_k = \frac{1}{4}$. As showed in Fig. 3, there exist directions of linear descent at zero. The claim follows. □

In fact, a direct calculation shows that $\hat{\partial} g(0) = \emptyset$ and g is smooth on $\mathbb{R}^n/\{0\}$, with $||\nabla g|| \geq 1$ everywhere, so, using the language of Definition 5.1, zero is not limiting stationary. However, zero is Clarke stationary.

For this example, the inexact-line-search BFGS method in exact arithmetic is thus known to converge to a point at which there exist directions of linear descent. However, numerical experiments reveal this convergence to be numerically unstable. The examples in [13] are complementary: convergence to points with directions of linear descent appears numerically stable but remains unproven in exact arithmetic.

6 Line-Search BFGS Versus Trust-Region BFGS

Given the apparent success of line-search BFGS methods on nonsmooth functions, it is natural to compare with trust-region versions. We consider here a trust-region BFGS algorithm from [11]. Given a starting point x_0, initial Hessian approximation B_0, trust-region radius Δ_0, maximum number of iterations N, parameters $\eta \in (0, 10^{-3})$ and $r \in (0, 1)$, the trust-region BFGS in this paper is as follows.

Algorithm (Trust-region BFGS algorithm)

$k \leftarrow 0$;

while $k < N$;

Exactly solve the subproblem

$$s_k \leftarrow \operatorname{argmin}\left\{\nabla f_k^T s + \frac{1}{2} s^T B_k s : \|s\| \leq \Delta_k\right\};$$

Compute

$$y_k \leftarrow \nabla f(x_k + s_k) - \nabla f_k$$
$$\text{ared} \leftarrow f_k - f(x_k + s_k)$$
$$\text{pred} \leftarrow -(\nabla f_k^T s_k + \frac{1}{2} s_k^T B_k s_k);$$

if $\frac{\text{ared}}{\text{pred}} > \eta$

$$x_{k+1} \leftarrow x_k + s_k;$$

else $x_k = x_{k+1}$;

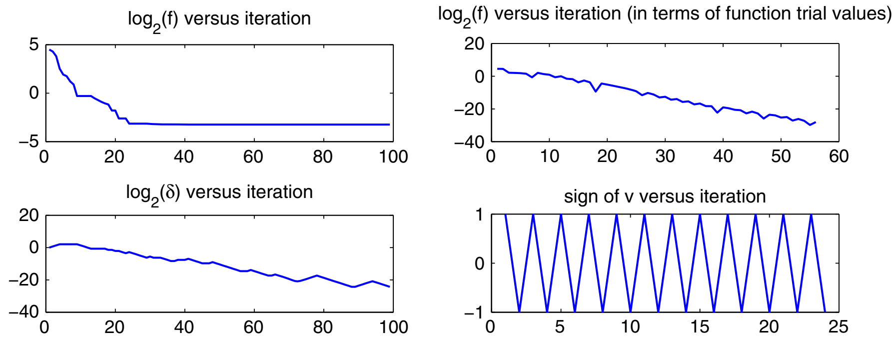
end(if)

if $\frac{\text{ared}}{\text{pred}} > 0.75$

if $||s_k|| \leq 0.8\Delta_k$, $\Delta_{k+1} \leftarrow \Delta_k$;

else $\Delta_{k+1} \leftarrow 2\Delta_k$;

end(if)



(a) trust region BFGS **(b)** line search BFGS

Fig. 4 Numerical results on $f([u\ v]^T) = u^2 + |v|$

elseif $0.1 \leq \frac{\text{ared}}{\text{pred}} \leq 0.75,\ \Delta_{k+1} =\leftarrow \Delta_k;$

else $\Delta_{k+1} = 0.5\Delta_k;$

end(if)

if $|s_k^T(y_k - B_k s_k)| \geq r||s_k|| \cdot ||y_k - B_k s_k||,$

$$B_{k+1} \leftarrow B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k};$$

else $B_{k+1} \leftarrow B_k;$

end(if) $k \leftarrow k + 1;$

end(while)

Numerical experiments show that line-search BFGS methods work well for broad classes of nonsmooth functions, while trust-region versions fail even on simple examples. In this section, we use the simple nonsmooth function $f([u\ v]^T) = u^2 + |v|$ to explore some intuitive reasons for the success of line-search BFGS methods over their trust-region counterparts.

We present some simple numerical experiments. The left graph in Fig. 4 is an example where trust-region BFGS fails to converge to the optimal solution. In contrast, the right graph in the same figure shows the success of the line-search BFGS on the same example.

Points on the line $v = 0$ are nonsmooth. Numerical results show that the line-search BFGS method generates a sequence of points that eventually cross that line at every iteration (see the lower right panel in the figure above). Indeed, this property can be proved analytically for the exact-line search BFGS, as we saw above. However, the trust-region BFGS method seems to satisfy no analogous property. The trust region seems overly restrictive on the updated point and approximate Hessian. In comparison, the line-search BFGS method responds much more effectively to the nonsmooth structure of the function.

Second, the line-search BFGS method updates the approximate Hessian when it finds a point satisfying the Armijo and Wolfe conditions along the current search direction, and these conditions seem to ensure that the updated point is satisfactory for this update. However, the trust-region BFGS updates the approximated Hessian matrix at each iteration, even when the current subproblem is not a good approximation of the original problem around the current point.

Third, numerical results show that the radius of the trust region converges to zero quickly (see the lower left figure above). When the trust region is small, the method cannot take a big step even though the subproblem is a good approximation of the original problem. This causes the method to converge very slowly. In addition, for the same reason, the method fails to take advantage of a well approximated subproblem to better update the approximate Hessian.

7 Conclusions

In practice, the BFGS algorithm seems a good general-purpose nonsmooth optimization method: it is relatively simple to implement, robust, and reliable. On the other hand, we are far from a complete theoretical understanding of its success. Our explorations in this work are first steps: two simple closed-form examples to help investigate how linear convergence occurs, a computational study of how well such examples might predict the relationship between convergence rate and problem conditioning (whatever that might be for this problem class), and a rigorous if rudimentary first step in a convergence proof for piecewise linear functions. While encouraging, the latter proof serves also to illustrate the difficulties in a careful analysis. A wider range of well-behaved examples, candidate condition numbers controlling convergence rate, and true convergence proofs would all constitute interesting advances in our understanding of this intriguing algorithm.

Acknowledgments The authors are grateful to Michael Overton for many helpful suggestions. The research was supported by National Science Foundation Grant DMS-0806057.

References

1. Powell, M.J.D.: Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In: Cottle, R., Lemke, R.W. (eds.) Nonlinear Programming. SIAM-AMS Proceedings 9, pp. 53–72. Providence (1976)
2. Powell, M.J.D.: Some properties of the variable metric algorithm. In: Lootsma, F.A. (ed.) Numerical Methods for Nonlinear Optimization, pp. 1–17. Academic Press, New York (1972)
3. Powell, M.J.D.: On the convergence of the variable metric algorithm. J. Inst. Math. Appl. **7**, 21–36 (1971)
4. 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)
5. Li, D.H., Fukushima, M.: A modified BFGS method and its global convergence in nonconvex minimization. J. Comput. Appl. Math. **129**, 15–35 (2001)
6. Mascarenhas, W.F.: The BFGS method with exact line searches fails for non-convex objective functions. Math. Program. **99**, 49–61 (2004)

7. Dai, Y.-H.: Convergence properties of the BFGS algorithm. SIAM J. Optim. **13**, 693–701 (2003)
8. Lemaréchal, C.: Numerical experiments in nonsmooth optimization. In: Nurminski, E.A. (ed.) Progress in Nondifferentiable Optimization, pp. 61–84. IIASA, Laxenburg (1982)
9. Lewis, A.S., Overton, M.L.: Nonsmooth optimization via quasi-Newton methods. Math. Program. **141**, 135–163 (2013)
10. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust Region Methods. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2000)
11. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (2006)
12. Lewis, A.S.: Active sets, nonsmoothness and sensitivity. SIAM J. Optim. **13**, 702–725 (2003)
13. Gürbüzbalaban, M., Overton, M.L.: On Nesterov's nonsmooth Chebyshev–Rosenbrock functions. Nonlinear Anal. Theory Methods Appl. **75**, 1282–1289 (2012)
14. Rockafellar, R.T., Wets, R.J.-B.: Variational Analysis. Springer, Berlin (1998)