A ROBUST GRADIENT SAMPLING ALGORITHM FOR NONSMOOTH, NONCONVEX OPTIMIZATION*

JAMES V. BURKE†, ADRIAN S. LEWIS‡, AND MICHAEL L. OVERTON§

Abstract. Let f be a continuous function on $\mathbf{R}^n$, and suppose f is continuously differentiable on an open dense subset. Such functions arise in many applications, and very often minimizers are points at which f is not differentiable. Of particular interest is the case where f is not convex, and perhaps not even locally Lipschitz, but is a function whose gradient is easily computed where it is defined. We present a practical, robust algorithm to locally minimize such functions, based on *gradient sampling*. No subgradient information is required by the algorithm.

When f is locally Lipschitz and has bounded level sets, and the sampling radius ϵ is fixed, we show that, with probability 1, the algorithm generates a sequence with a cluster point that is Clarke ϵ-stationary. Furthermore, we show that if f has a unique Clarke stationary point $\bar{x}$, then the set of all cluster points generated by the algorithm converges to $\bar{x}$ as ϵ is reduced to zero.

Numerical results are presented demonstrating the robustness of the algorithm and its applicability in a wide variety of contexts, including cases where f is not locally Lipschitz at minimizers. We report approximate local minimizers for functions in the applications literature which have not, to our knowledge, been obtained previously. When the termination criteria of the algorithm are satisfied, a precise statement about nearness to Clarke ϵ-stationarity is available. A MATLAB implementation of the algorithm is posted at http://www.cs.nyu.edu/overton/papers/gradsamp/alg.

1. Introduction. The analysis of nonsmooth, nonconvex functions has been a rich area of mathematical research for three decades. Clarke introduced the notion of generalized gradient in [Cla73, Cla83]; comprehensive studies of more recent developments may be found in [CLSW98, RW98]. The generalized gradient of a function f at a point x reduces to the gradient if f is smooth at x and to the subdifferential if f is convex; hence, we follow common usage in referring to the generalized gradient as the (Clarke) subdifferential, or set of (Clarke) subgradients. Its use in optimization algorithms began soon after its appearance in the literature. In particular, the concept of the ϵ-steepest descent direction for locally Lipschitz functions was introduced by Goldstein in [Gol77]; another early paper is [CG78]. It is well known that the ordinary steepest descent algorithm typically fails by converging to a nonoptimal point when applied to nonsmooth functions, whether convex or not. The fundamental difficulty is that most interesting nonsmooth objective functions have minimizers where the gradient is not defined.

An extensive discussion of several classes of algorithms for the minimization of

---

nonsmooth, nonconvex, locally Lipschitz functions, complete with convergence analysis, may be found in Kiwiel's book [Kiw85]. What these algorithms have in common is that, at each iteration, they require the computation of a single subgradient (not the entire subdifferential set) in addition to the value of the function. The algorithms then build up information about the subdifferential properties of the function using ideas known as bundling and aggregation. Such "bundle" algorithms, as they are generally known, are especially effective for nonsmooth, convex optimization because of the global nature of convexity, and the ideas in this case trace back to [Lem75, Wol75]; for a comprehensive discussion of the convex case, see [HUL93]. However, for nonconvex functions, subgradient information is meaningful only locally and must be discounted when no longer relevant. The consequence is that bundle algorithms are necessarily much more complicated in the nonconvex case. Other contributions to nonconvex bundle methods since Kiwiel's book was published in 1985 include [FGG02, Gro02, LSB91, LV98, MN92, OKZ98, SZ92]. Despite this activity in the field, the only publicly available nonconvex bundle software of which we are aware are the Bundle Trust (BT) FORTRAN code dating from 1991 [SZ92] and some more recent FORTRAN codes of [LV98].

In addition to this body of work on general nonsmooth, nonconvex optimization, there is a large literature on more specialized problems, including nonconvex polyhedral functions [Osb85], compositions of convex and smooth functions [Bur85, Fle87], and quasi-differentiable functions [DR95].

There are many reasons why most algorithms for nonsmooth optimization do not ask the user to provide a description of the entire subdifferential set at each iterate. One is that this would demand a great deal of the user in all but the simplest applications. More fundamentally, it is not clear how one would represent such a set in general since it is already a formidable task in the polyhedral setting [Osb85]. Even if this were resolved, implementation would be difficult for the user given the inherent complexity of the continuity properties of these set-valued mappings. Asking the user to provide only one subgradient at a point resolves these difficulties.

In virtually all interesting applications, the function being minimized is continuously differentiable almost everywhere, although it is often not differentiable at minimizers. Under this assumption, when a user is asked to provide a subgradient at a randomly selected point, with probability 1 the subgradient is unique, namely, the gradient. This observation led us to consider a simple gradient sampling algorithm, first presented without any analysis in [BLO02b]. At a given iterate, we compute the gradient of the objective function on a set of randomly generated nearby points, and use this information to construct a local search direction that may be viewed as an approximate ϵ-steepest descent direction, where ϵ is the sampling radius. As is standard for algorithms based on subgradients, we obtain the descent direction by solving a quadratic program. Gradient information is not saved from one iteration to the next, but discarded once a lower point is obtained from a line search. A key motivating factor is that, in many applications, computing the gradient when it exists is little additional work once the function value is computed. Often, well-known formulas for the gradient are available; alternatively, automatic differentiation might be used. No subgradient information is required from the user. We have found the gradient sampling algorithm to be very effective for approximating local minimizers of a wide variety of nonsmooth, nonconvex functions, including non-Lipschitz functions.

In a separate work [BLO02a], we analyzed the extent to which the Clarke subdifferential at a point can be approximated by random sampling of gradients at nearby

points, justifying the notion that the convex hull of the latter set can serve as a surrogate for the former.

This paper is organized as follows. The gradient sampling (GS) algorithm is presented in section 2. The sampling radius ϵ may be fixed for all iterates or may be reduced dynamically; this is controlled by the choice of parameters defining the algorithm.

A convergence analysis is given in section 3, making the assumption that the function $f : \mathbf{R}^n \to \mathbf{R}$ is locally Lipschitz, has bounded level sets, and, in addition, is continuously differentiable on an open dense subset of $\mathbf{R}^n$. Our first convergence result analyzes the GS algorithm with fixed ϵ, and establishes that, with probability 1, it generates a sequence with a cluster point that is Clarke ϵ-stationary, in a sense that will be made precise. A corollary shows that if f has a unique Clarke stationary point $\bar{x}$, then the sets of all cluster points generated by the GS algorithm converge to $\bar{x}$ as ϵ is reduced to zero. These results are then strengthened for the case where f is either convex or smooth. In all cases, when the termination criteria of the GS algorithm are satisfied, a precise statement about nearness to Clarke ϵ-stationarity is available.

We should emphasize that although Clarke stationarity is a first-order optimality condition, there are two considerations that allow us to expect that, in practice, cluster points of the GS algorithm are more than just approximate stationary points, but are in fact approximate local minimizers. The first consideration is a very practical one: the line search enforces a descent property for the sequence of iterates. The second consideration is more theoretical: we are generally interested in applying the algorithm to a nonsmooth function that, although not convex, is *subdifferentially regular* [RW98] (equivalently, its epigraph is regular in the sense of Clarke [Cla83]). Clarke stationarity at a point of subdifferential regularity implies the nonnegativity of the usual directional derivative in all directions. This is much stronger than Clarke stationarity in the absence of regularity. For example, 0 is a Clarke stationary point of the function $f(x) = -|x|$, but f is not subdifferentially regular at 0.

In section 4, we present numerical results that demonstrate the effectiveness and robustness of the GS algorithm and its applicability in a variety of contexts. We set the parameters defining the GS algorithm so that the sampling radius ϵ is reduced dynamically. We begin with a classical problem: Chebyshev exponential approximation. Our second example involves minimizing a product of eigenvalues of a symmetric matrix; it arises in an environmental data analysis application. We then turn to some important functions arising in nonsymmetric matrix analysis and robust control, including non-Lipschitz spectral functions, pseudospectral functions, and the distance to instability. We conclude with a stabilization problem for a model of a Boeing 767 at a flutter condition. As far as we know, none of the problems that we present has been solved previously by any method.

Finally, we make some concluding remarks in section 5. Our MATLAB implementation of the GS algorithm is freely available on the Web.[1]

2. The gradient sampling algorithm. The GS algorithm is conceptually very simple. Basically, it is a stabilized steepest descent algorithm. At each iteration, a descent direction is obtained by evaluating the gradient at the current iterate and at additional nearby points and then computing the vector in the convex hull of these gradients with smallest norm. A standard line search is then used to obtain a lower

---

[1] http://www.cs.nyu.edu/overton/papers/gradsamp/alg/.

point. Thus, stabilization is controlled by the sampling radius used to sample the gradients. In practice, we begin with a large sampling radius and then reduce this according to rules that are set out in section 4, where we show, using various examples, how well the algorithm works.

Despite the simplicity and power of the algorithm, its analysis is not so simple. One difficulty is that it is inherently probabilistic, since the gradient is not defined on the whole space. Another is that analyzing its convergence for a fixed sampling radius is already challenging, and extending our results in that case to a version of the algorithm that reduces the sampling radius dynamically presents additional difficulties. In order to take care of both fixed and dynamically changing sampling radius, the statement of the algorithm becomes a little more complicated. We set out the algorithmic details in this section and present the convergence analysis in the next section.

The algorithm may be applied to any function $f : \mathbf{R}^n \to \mathbf{R}$ that is continuous on $\mathbf{R}^n$ and differentiable almost everywhere. However, all our theoretical results assume that f is locally Lipschitz continuous and continuously differentiable on an open dense subset D of $\mathbf{R}^n$. In addition, we assume that there is a point $\tilde{x} \in \mathbf{R}^n$ for which the set $\mathcal{L} = \{x \mid f(x) \leq f(\tilde{x})\}$ is compact.

The local Lipschitz hypothesis allows us to approximate the Clarke subdifferential [Cla83] as follows. For each $\epsilon > 0$, define the multifunction $G_\epsilon : \mathbf{R}^n \rightrightarrows \mathbf{R}^n$ by

$$G_\epsilon(x) = \operatorname{cl}\operatorname{conv}\nabla f((x + \epsilon\mathbb{B}) \cap D),$$

where $\mathbb{B} = \{x \mid \|x\| \leq 1\}$ is the closed unit ball and $\|\cdot\|$ is the 2-norm. The sets $G_\epsilon(x)$ can be used to give the following representation of the Clarke subdifferential of f at a point x:

$$\bar{\partial}f(x) = \bigcap_{\epsilon > 0} G_\epsilon(x).$$

We also make use of the ϵ-subdifferential introduced by Goldstein [Gol77]. For each $\epsilon > 0$, the Clarke ϵ-subdifferential is given by

$$\bar{\partial}_\epsilon f(x) = \operatorname{cl}\operatorname{conv}\bar{\partial}f(x + \epsilon\mathbb{B}).$$

Clearly, $G_\epsilon(x) \subset \bar{\partial}_\epsilon f(x)$, and for $0 < \epsilon_1 < \epsilon_2$ we have $\bar{\partial}_{\epsilon_1} f(x) \subset G_{\epsilon_2}(x)$. In addition, it is easily shown that the multifunction $\bar{\partial}_\epsilon f$ has closed graph.

We say that a point x is a Clarke ϵ-stationary point for f if $0 \in \bar{\partial}_\epsilon f(x)$. This notion of ϵ-stationarity is key to our approach. Indeed, the algorithm described below is designed to locate Clarke ϵ-stationary points. For this reason we introduce the following scalar measure of proximity to Clarke ϵ-stationarity:

$$(1) \qquad\qquad \rho_\epsilon(x) = \operatorname{dist}\left(0 \mid G_\epsilon(x)\right).$$

We now state the GS algorithm. Scalar parameters are denoted by lowercase Greek letters. A superscript on a scalar parameter indicates taking that scalar to the power of the superscript.

In order to facilitate the reading and analysis of the algorithm, we provide a partial glossary of the notation used in its statement.

Glossary of Notation	
k: Iteration counter.	μ: Sampling radius reduction factor.
x^k: Current iterate.	θ: Optimality tolerance reduction factor.
γ: Backtracking reduction factor.	m: Sample size.
$\mathcal{L}$: $\{x \mid f(x) \le f(\tilde{x})\}$.	D: Points of differentiability.
u^{kj}: Unit ball samples.	x^{kj}: Sampling points.
β: Armijo parameter.	g^k: Shortest approximate subgradient.
ϵ_k: Sampling radius.	d^k: Search direction.
ν_k: Optimality tolerance.	t_k: Step length.

The GS algorithm.

Step 0: (Initialization)

Let $x^0 \in \mathcal{L} \cap D$, $\gamma \in (0,1)$, $\beta \in (0,1)$, $\epsilon_0 > 0$, $\nu_0 \ge 0$, $\mu \in (0,1]$, $\theta \in (0,1]$, $k = 0$, and $m \in \{n+1, n+2, \dots\}$.

Step 1: (Approximate the Clarke ϵ-subdifferential by gradient sampling)

Let $u^{k1}, \dots, u^{km}$ be sampled independently and uniformly from $\mathbb{B}$, and set

$$x^{k0} = x^k \quad \text{and} \quad x^{kj} = x^k + \epsilon_k u^{kj}, \; j = 1, \dots, m.$$

If for some $j = 1, \dots, m$ the point $x^{kj} \notin D$, then STOP; otherwise, set

$$G_k = \text{conv}\{\nabla f(x^{k0}), \nabla f(x^{k1}), \dots, \nabla f(x^{km})\},$$

and go to Step 2.

Step 2: (Compute a search direction)

Let $g^k \in G_k$ solve the quadratic program $\min_{g \in G_k} \|g\|^2$, i.e.,

$$\|g^k\| = \text{dist}(0 \mid G_k) \quad \text{and} \quad g^k \in G_k.$$

If $\nu_k = \|g^k\| = 0$, STOP. If $\|g^k\| \le \nu_k$, set $t_k = 0$, $\nu_{k+1} = \theta\nu_k$, and $\epsilon_{k+1} = \mu\epsilon_k$, and go to Step 4; otherwise, set $\nu_{k+1} = \nu_k$, $\epsilon_{k+1} = \epsilon_k$, and $d^k = -g^k/\|g^k\|$, and go to Step 3.

Step 3: (Compute a step length)

Set

$$\begin{aligned} t_k = \quad &\max \; \gamma^s \\ &\text{subject to} \quad s \in \{0, 1, 2, \dots\} \text{ and} \\ &\qquad\qquad f(x^k + \gamma^s d^k) < f(x^k) - \beta\gamma^s\|g^k\|, \end{aligned}$$

and go to Step 4.

Step 4: (Update)

If $x^k + t_k d^k \in D$, set $x^{k+1} = x^k + t_k d^k$, $k = k + 1$, and go to Step 1. If $x^k + t_k d^k \notin D$, let $\hat{x}^k$ be any point in $x^k + \epsilon_k \mathbb{B}$ satisfying $\hat{x}^k + t_k d^k \in D$ and

$$(2) \qquad\qquad f(\hat{x}^k + t_k d^k) < f(x^k) - \beta t_k\|g^k\|$$

(such an $\hat{x}^k$ exists due to the continuity of f). Then set $x^{k+1} = \hat{x}^k + t_k d^k$, $k = k + 1$, and go to Step 1.

The algorithm is designed so that every iterate x^k is an element of the set $\mathcal{L} \cap D$. We now show that the line search defined in Step 3 of the algorithm is well defined in the sense that the value of t_k can be determined by a finite process. Let $N_C(x)$ denote the normal cone to the set C at a point x [Roc70], and recall from convex analysis

that g^k solves $\inf_{g \in G_k} \|g\|$ if and only if $-g^k \in N_{G_k}(g^k)$, that is, $\langle g - g^k, -g^k \rangle \leq 0$ for all $g \in G_k$. Therefore, if $\|g^k\| = \text{dist}\,(0 \mid G_k) \neq 0$, then

$$\nabla f(x^k)^T d^k \leq \sup_{g \in G_k} \langle g, d^k \rangle \leq -\|g^k\|.$$

Since $x^k \in D$, we have $f'(x^k; d^k) = \nabla f(x^k)^T d^k$. Hence, there is a $\bar{t} > 0$ such that

$$f(x^k + td^k) \leq f(x^k) + t\beta \nabla f(x^k)^T d^k \leq f(x^k) - t\beta \|g^k\| \quad \forall\, t \in (0, \bar{t}).$$

The choice of search direction used in the GS algorithm is motivated by the direction of steepest descent in nonsmooth optimization. Recall that the Clarke directional derivative for f at a point x is given by the support functional for the Clarke subdifferential at x:

$$f^\circ(x; d) = \max_{z \in \bar{\partial} f(x)} \langle z, d \rangle.$$

Therefore, the direction of steepest descent is obtained by solving the problem

$$\min_{\|d\| \leq 1} f^\circ(x; d) = \min_{\|d\| \leq 1} \max_{z \in \bar{\partial} f(x)} \langle z, d \rangle.$$

The next lemma, which is essentially well known, shows that the search direction in the GS algorithm is an approximate direction of steepest descent.

LEMMA 2.1. *Let G be any compact convex subset of $\mathbf{R}^n$; then*

$$(3) \qquad\qquad -\text{dist}\,(0 \mid G) = \min_{\|d\| \leq 1} \max_{g \in G} \langle g, d \rangle.$$

Moreover, if $\bar{g} \in G$ satisfies $\|\bar{g}\| = \text{dist}\,(0 \mid G)$, then $\bar{d} = -\bar{g}/\|\bar{g}\|$ solves the problem on the right-hand side of (3).

Proof. The result is an elementary consequence of the von Neumann minimax theorem. Indeed, one has

$$-\text{dist}\,(0 \mid G) = -\min_{g \in G} \|g\|$$

$$= -\min_{g \in G} \max_{\|d\| \leq 1} \langle g, d \rangle$$

$$= -\max_{\|d\| \leq 1} \min_{g \in G} \langle g, d \rangle$$

$$= -\max_{\|d\| \leq 1} \min_{g \in G} \langle g, -d \rangle$$

$$= \min_{\|d\| \leq 1} \max_{g \in G} \langle g, d \rangle,$$

from which it easily follows that $d = -\bar{g}/\|\bar{g}\|$ solves the problem

$$\inf_{\|d\| \leq 1} \sup_{g \in G} \langle g, d \rangle,$$

where $\bar{g}$ is the least norm element of G.   □

By setting G equal to the sets G_k in the GS algorithm, we obtain the *approximate steepest descent property*:

$$-\text{dist}\,(0 \mid G_k) = \min_{\|d\| \leq 1} \max_{g \in G_k} \langle g, d \rangle.$$

The case $x^k + t_k d^k \notin D$ in Step 4 of the algorithm seems unlikely to occur, and we do not correct for this possibility in our numerical implementation. Nonetheless, we need to compensate for it in our theoretical analysis since we have not been able to show that it is a zero probability event. The GS algorithm is a nondeterministic algorithm, and in this spirit Step 4 is easily implemented in a nondeterministic fashion as follows. At step $\omega = 1, 2, \dots$, sample $\hat{x}$ from a uniform distribution on $x^k + (\epsilon_k/\omega)\mathbb{B}$ and check to see if $\hat{x} + t_k d^k \in D$ and the inequality (2) with $\hat{x}^k = \hat{x}$ is satisfied. If so, set $\hat{x}^k = \hat{x}$, $x^{k+1} = \hat{x}^k + t_k d^k$, $k = k + 1$, and return to Step 1; otherwise, increase ω and repeat. With probability 1 this procedure terminates finitely.

The GS algorithm can be run with $\nu_0 = 0$ and $\mu = 1$, so that $\nu_k = 0$ and $\epsilon_k = \epsilon_0$ for all k. This instance of the algorithm plays a prominent role in our convergence analysis. Indeed, all of our theoretical convergence results follow from the analysis in this case. In practice, however, the algorithm is best implemented with $\mu < 1$ and ν_0 positive. When $\nu_0 = 0$, the algorithm terminates at iteration k_0 if either $x^{k_0 j} \notin D$ for some $j = 1, \dots, m$ or $\left\| g^{k_0} \right\| = 0$. The probability that $x^{k_0 j} \notin D$ for some $j = 1, \dots, m$ is zero, while $\left\| g^{k_0} \right\| = 0$ is equivalent to $\rho_{\epsilon_{k_0}}(x^{k_0}) = 0$.

Before proceeding to the convergence analysis, we make a final observation concerning the stochastic structure of the algorithm, as it plays a key role in our analysis. Although the algorithm specifies that the points $u^{k1}, \dots, u^{km}$ are sampled from $\mathbb{B}$ at each iteration, we may think of this sequence as a realization of a stochastic process $\{(\mathbf{u}^{k1}, \dots, \mathbf{u}^{km})\}$ where the realization occurs before the initiation of the algorithm. In this regard, we consider only those realizations that are ergodic with respect to $\mathbb{B}^m$. Specifically, we consider only those processes that hit every positive measure subset of $\mathbb{B}^m$ infinitely often. We define this subset of events as $\mathcal{E}$ and note that with probability 1 the realization $\{(u^{k1}, \dots, u^{km})\}$ is in $\mathcal{E}$.

3. Convergence analysis. Throughout this section it is assumed that the function $f : \mathbf{R}^n \to \mathbf{R}$ is locally Lipschitz continuous on $\mathbf{R}^n$ and continuously differentiable on an open dense subset D of $\mathbf{R}^n$. We begin with two technical lemmas.

LEMMA 3.1. *Let $v \in C$, where C is a nonempty closed convex subset of $\mathbf{R}^n$ that does not contain the origin. If $\delta > 0$, $\eta > 0$, and $u, \bar{u} \in C$ are such that $\eta \le \|\bar{u}\| = \text{dist}\,(0 \mid C)$ and $\|u\| \le \|\bar{u}\| + \delta$, then*

$$\left\langle v - u\,, \frac{-u}{\|u\|} \right\rangle \le \left[\|v\| \sqrt{\frac{2}{\eta}} + \sqrt{[2\,\|\bar{u}\| + \delta]} \right] \sqrt{\delta}.$$

Proof. Since $\|\bar{u}\| = \text{dist}\,(0 \mid C)$, we have $-\bar{u} \in N_C(\bar{u})$. Hence, for all $h \in C$

$$\langle h - \bar{u}\,, -\bar{u} \rangle \le 0,$$

or, equivalently,

$$\|\bar{u}\|^2 \le \langle h\,, \bar{u} \rangle.$$

Therefore,

$$1 - \left\langle \frac{u}{\|u\|}\,, \frac{\bar{u}}{\|\bar{u}\|} \right\rangle \le 1 - \frac{\|\bar{u}\|^2}{\|u\|\,\|\bar{u}\|} = \frac{1}{\|u\|}[\|u\| - \|\bar{u}\|] \le \frac{\delta}{\|u\|} \le \frac{\delta}{\|\bar{u}\|},$$

$$\left\| \frac{u}{\|u\|} - \frac{\bar{u}}{\|\bar{u}\|} \right\|^2 = 2 \left[1 - \left\langle \frac{u}{\|u\|}\,, \frac{\bar{u}}{\|\bar{u}\|} \right\rangle \right] \le \frac{2\delta}{\|\bar{u}\|},$$

$$\|u - \bar{u}\|^2 = \|u\|^2 - 2\langle u, \bar{u}\rangle + \|\bar{u}\|^2$$
$$\leq \|u\|^2 - \|\bar{u}\|^2$$
$$= (\|u\| + \|\bar{u}\|)(\|u\| - \|\bar{u}\|)$$
$$\leq [2\|\bar{u}\| + \delta]\delta,$$

and

$$\|v - \bar{u}\|^2 = \|v\|^2 - 2\langle v, \bar{u}\rangle + \|\bar{u}\|^2$$
$$\leq \|v\|^2 - \|\bar{u}\|^2$$
$$\leq \|v\|^2.$$

Consequently,

$$\left\langle v - u, \frac{-u}{\|u\|} \right\rangle = \left\langle v - \bar{u}, \frac{-\bar{u}}{\|\bar{u}\|} \right\rangle + \left\langle v - \bar{u}, \frac{\bar{u}}{\|\bar{u}\|} - \frac{u}{\|u\|} \right\rangle + \left\langle u - \bar{u}, \frac{u}{\|u\|} \right\rangle$$
$$\leq \left\langle v - \bar{u}, \frac{\bar{u}}{\|\bar{u}\|} - \frac{u}{\|u\|} \right\rangle + \left\langle u - \bar{u}, \frac{u}{\|u\|} \right\rangle$$
$$\leq \|v - \bar{u}\| \left\| \frac{u}{\|u\|} - \frac{\bar{u}}{\|\bar{u}\|} \right\| + \|u - \bar{u}\|$$
$$\leq \|v\| \sqrt{\frac{2\delta}{\|\bar{u}\|}} + \sqrt{[2\|\bar{u}\| + \delta]\delta}$$
$$\leq \left[\|v\| \sqrt{\frac{2}{\eta}} + \sqrt{[2\|\bar{u}\| + \delta]} \right] \sqrt{\delta}. \qquad \square$$

The next lemma establishes properties of the set of all points close to a given point x' that can be used to provide a δ-approximation to the element of $G_\epsilon(x')$ of least norm. Specifically, let $m \geq n + 1$, $\delta > 0$, and $x', x \in \mathbf{R}^n$ be given, and set

$$D_\epsilon^m(x) = \prod_1^m (D \cap (x + \epsilon\mathbb{B})) \subset \prod_1^m \mathbf{R}^n.$$

Consider the set $R_\epsilon(x', x, \delta) \subset \prod_1^{m+1} \mathbf{R}^n$ of all $(m+1)$-tuples $(x^1, \ldots, x^m, g)$ satisfying

$$(x^1, \ldots, x^m) \in D_\epsilon^m(x) \quad \text{and} \quad g = \sum_{j=1}^m \lambda_j \nabla f(x^j)$$

for some $0 \leq \lambda_j$, $j = 1, 2, \ldots, m$, with

$$\sum_{j=1}^m \lambda_j = 1 \quad \text{and} \quad \left\| \sum_{j=1}^m \lambda_j \nabla f(x^j) \right\| \leq \rho_\epsilon(x') + \delta.$$

We need to understand the local behavior of this set as well as its projections

$$V_\epsilon(x', x, \delta) = \left\{ (x^1, \ldots, x^m) : \exists g \text{ with } (x^1, \ldots, x^m, g) \in R_\epsilon(x', x, \delta) \right\}$$

and

$$U_\epsilon(x', x, \delta) = \left\{ g : \exists (x^1, \ldots, x^m) \text{ with } (x^1, \ldots, x^m, g) \in R_\epsilon(x', x, \delta) \right\}.$$

LEMMA 3.2. *For $\epsilon > 0$, let ρ_ϵ be as defined in* (1), *and let $m \geq n+1$, $\delta > 0$, and $\bar{x} \in \mathbf{R}^n$ be given with $0 < \rho_\epsilon(\bar{x})$.*

(i) *There is a $\tau > 0$ such that the set $V_\epsilon(\bar{x}, x, \delta)$ contains a nonempty open subset whenever $\|x - \bar{x}\| \leq \tau$.*

(ii) *$\tau > 0$ may be chosen so that there is a nonempty open set $\bar{V} \subset V_\epsilon(\bar{x}, \bar{x}, \delta)$ such that $\bar{V} \subset V_\epsilon(\bar{x}, x, \delta)$ for all $x \in \bar{x} + \tau\mathbb{B}$.*

(iii) *By definition, we have $U_\epsilon(\bar{x}, x, \delta) \subset G_\epsilon(x)$ for all $x \in \mathbf{R}^n$, and so*

$$\rho_\epsilon(x) \leq \|u\| \leq \rho_\epsilon(\bar{x}) + \delta \quad \forall\, u \in U_\epsilon(\bar{x}, x, \delta),\ x \in \mathbf{R}^n.$$

In addition, if $\tau > 0$ is as given by statement (i) *or* (ii)*, then the set $U_\epsilon(\bar{x}, x, \delta)$ is guaranteed to be nonempty whenever $\|x - \bar{x}\| \leq \tau$.*

(iv) *The function ρ_ϵ is upper semicontinuous, i.e.,*

$$\limsup_{x \to \bar{x}} \rho_\epsilon(x) \leq \rho_\epsilon(\bar{x}).$$

(v) *Let $\eta > 0$. Then for every compact subset K of the set $\mathcal{L}$ for which $\inf_K \rho_\epsilon(x) \geq \eta$ there exist $\bar{\mu} > 0$, an integer $\ell \geq 1$, $\tau_j \in (0, \epsilon/3)$, $j = 1, \ldots, \ell$, and a set of points $\{z^1, z^2, \ldots, z^\ell\} \subset K$ such that the union $\cup_{j=1}^\ell (z^j + \tau_j \operatorname{int} \mathbb{B})$ is an open cover of K and for each $j = 1, \ldots, \ell$ there is a point $(z^{j1}, \ldots, z^{jm}) \in V_\epsilon(z^j, z^j, \delta)$ such that*

$$(z^{j1}, \ldots, z^{jm}) + \bar{\mu}\mathbb{B}^m \subset V_\epsilon(z^j, x, \delta) \text{ whenever } x \in z^j + \tau_j\mathbb{B}.$$

Proof. Let $u \in \operatorname{conv} \{\nabla f(x) \mid x \in (\bar{x} + \epsilon\mathbb{B}) \cap D\}$ be such that

$$\|u\| < \rho_\epsilon(\bar{x}) + \delta.$$

Then Carathéodory's theorem [Roc70] implies the existence of $(\bar{x}^1, \ldots, \bar{x}^m) \in D_\epsilon^m(\bar{x})$ and $\bar{\lambda} \in \mathbf{R}_+^m$ with $\sum_{j=1}^m \bar{\lambda}_j = 1$ such that $u = \sum_{j=1}^m \bar{\lambda}_j \nabla f(\bar{x}^j)$. Since f is continuously differentiable on the open set D, there is an $\epsilon_0 > 0$ such that f is continuously differentiable on $\bar{x}^j + \epsilon_0 \operatorname{int} \mathbb{B} \subset \bar{x} + \epsilon\mathbb{B}$ for $j = 1, \ldots, m$. Define $F : (\bar{x}^1 + \epsilon_0 \operatorname{int} \mathbb{B}) \times \cdots \times (\bar{x}^m + \epsilon_0 \operatorname{int} \mathbb{B}) \to \mathbf{R}^n$ by $F(x^1, \ldots, x^m) = \sum_{j=1}^m \bar{\lambda}_j \nabla f(x^j)$. The mapping F is continuous on $(\bar{x}^1 + \epsilon_0 \operatorname{int} \mathbb{B}) \times \cdots \times (\bar{x}^m + \epsilon_0 \operatorname{int} \mathbb{B})$. Next define $U = \{u \in \mathbf{R}^n \mid \|u\| < \rho_\epsilon(\bar{x}) + \delta\}$. Then, by definition, the set

$$V = F^{-1}(U) \cap \left((\bar{x}^1 + \epsilon_0 \operatorname{int} \mathbb{B}) \times \cdots \times (\bar{x}^m + \epsilon_0 \operatorname{int} \mathbb{B})\right)$$

is a nonempty open subset of $V_\epsilon(\bar{x}, \bar{x}, \delta)$. Now since the sets $x + \epsilon\mathbb{B}$ converge to the set $\bar{x} + \epsilon\mathbb{B}$ in the Hausdorff metric as $x \to \bar{x}$, we must have that $V \cap ((x + \epsilon \operatorname{int} \mathbb{B}) \times \cdots \times (x + \epsilon \operatorname{int} \mathbb{B}))$ is open and nonempty for all x sufficiently close to $\bar{x}$. This proves statement (i).

To see that statement (ii) is true, observe that since $V_\epsilon(\bar{x}, \bar{x}, \delta)$ contains a nonempty open subset $\tilde{V}$ there must exist an $\tilde{\epsilon} \in (0, \epsilon)$ such that

$$\bar{V} = \tilde{V} \cap ((\bar{x} + \tilde{\epsilon} \operatorname{int} \mathbb{B}) \times \cdots \times (\bar{x} + \tilde{\epsilon} \operatorname{int} \mathbb{B}))$$

is open and nonempty. Since the Hausdorff distance between $x + \epsilon\mathbb{B}$ and $\bar{x} + \epsilon\mathbb{B}$ is $\|x - \bar{x}\|$ whenever $\|x - \bar{x}\| < \epsilon/2$, we have that $\bar{V} \subset V_\epsilon(\bar{x}, x, \delta)$ whenever $\|x - \bar{x}\| \leq (\epsilon - \tilde{\epsilon})/2 = \tau$. This proves statement (ii).

Statement (iii) follows immediately from statement (ii), while statement (iv) follows from statement (iii) by letting $\delta \to 0$.

Statement (ii) and compactness imply statement (v). Indeed, statement (ii) implies that for each $x' \in K$ there is a $\tau(x') \in (0, \epsilon/3)$ and a nonempty open set

$V(x') \subset V_\epsilon(x', x', \delta)$ such that $V(x') \subset V_\epsilon(x', x, \delta)$ whenever $x \in x' + \tau(x')\mathbb{B}$. The sets $x' + \tau(x') \operatorname{int}\mathbb{B}$ form an open cover of K. Since K is compact, this open cover contains a finite subcover $z^j + \tau_j \operatorname{int}\mathbb{B}, j = 1, \ldots, \ell$. Let $(z^{j1}, \ldots, z^{jm}) \in V_\epsilon(z^j, z^j, \delta)$ and $\bar{\mu}_j > 0$ be such that $(z^{j1}, \ldots, z^{jm}) + \bar{\mu}_j \mathbb{B}^m \subset V(z^j)$ for $j = 1, \ldots, \ell$. By setting $\bar{\mu} = \min\{\bar{\mu}_1, \ldots, \bar{\mu}_\ell\}$ we obtain the result. $\quad\square$

We also need the following mean value inequality [Cla83, Theorem 2.3.7].

THEOREM 3.3 (Lebourg mean value theorem). *Let $f : \mathbf{R}^n \to \mathbf{R}$ be locally Lipschitz and let $x, y \in \mathbf{R}^n$. Then there exist $z \in [x, y]$ and $w \in \bar{\partial}f(z)$ such that*

$$f(y) - f(x) = \langle w, y - x \rangle.$$

The main convergence result follows.

THEOREM 3.4 (convergence for fixed sampling radius). *If $\{x^k\}$ is a sequence generated by the GS algorithm with $\epsilon_0 = \epsilon$, $\nu_0 = 0$, and $\mu = 1$, then with probability 1 either the algorithm terminates finitely at some iteration k_0 with $\rho_\epsilon(x^{k_0}) = 0$ or there is a subsequence $J \subset \mathbb{N}$ such that $\rho_\epsilon(x^k) \to_J 0$ and every cluster point $\bar{x}$ of the subsequence $\{x^k\}_J$ satisfies $0 \in \bar{\partial}_\epsilon f(\bar{x})$.*

Proof. We may assume that event $\mathcal{E}$ occurs (see the discussion at the end of section 2). That is, we may assume that the sequence $\{(u^{k1}, \ldots, u^{km})\}$ hits every positive measure subset of $\mathbb{B}^m$ infinitely often. As previously noted, this event occurs with probability 1.

We begin by considering the case where the algorithm terminates finitely. Let $x \in \mathcal{L}$ and $\epsilon > 0$, and let z be a realization of a random variable that is uniformly distributed on $\mathbb{B}$. Then the probability that $x + \epsilon z \notin D$ is zero. Hence, with probability 1 the algorithm does not terminate in Step 1. Therefore, if the algorithm terminates finitely at some iteration k_0, then with probability 1 it did so in Step 2 with $\rho_\epsilon(x^{k_0}) = 0$.

We now restrict our attention to the set of events $\hat{\mathcal{E}} \subset \mathcal{E}$ where the algorithm does not terminate finitely. For such events, we have that $x^{kj} \in D$ for $j = 0, 1, \ldots, m$ and $k = 0, 1, 2, \ldots$. Conditioned on $\hat{\mathcal{E}}$ occurring, we show that with probability 1 there is a subsequence $J \subset \mathbb{N}$ such that $\rho_\epsilon(x^k) \to_J 0$ and every cluster point $\bar{x}$ of the subsequence $\{x^k\}_J$ satisfies $0 \in \bar{\partial}_\epsilon f(\bar{x})$.

Since the sequence $\{f(x^k)\}$ is decreasing and $\mathcal{L}$ is compact, it must be the case that there is a $\kappa > 0$ and an $\hat{f}$ such that κ is a Lipschitz constant for f on all of $\mathcal{L} + \mathbb{B}$ and $f(x^k) \downarrow \hat{f}$. Consequently, $(f(x^{k+1}) - f(x^k)) \to 0$, and so by Step 3 of the GS algorithm

$$t_k \left\| g^k \right\| \to 0.$$

If the result were false, then with positive probability there is an $\eta > 0$ such that

$$\eta = \inf_{k \in \mathbb{N}} \rho_\epsilon(x^k) \leq \inf_{k \in \mathbb{N}} \left\| g^k \right\|, \tag{4}$$

since by definition $\rho_\epsilon(x) = \operatorname{dist}(0 \mid G_\epsilon(x))$ and $\left\| g^k \right\| = \operatorname{dist}(0 \mid G_k)$. For such an event $t_k \downarrow 0$.

Let us assume that event (4) has occurred, and let K be the set of points $x \in \mathcal{L}$ having $\rho_\epsilon(x) \geq \eta$. Then

$$\{x^k\} \subset K. \tag{5}$$

The set K is closed by part (iv) of Lemma 3.2 and so K is compact since $\mathcal{L}$ is compact. Choose $\delta > 0$ so that

$$\text{(6)} \qquad \left[\kappa\sqrt{\frac{2}{\eta}} + \sqrt{2\kappa + \delta} \right] \sqrt{\delta} < (1 - \beta)\frac{\eta}{2}.$$

Let $\bar{\mu} > 0$, $\tau_j \in (0, \epsilon/3)$, $z^j \in K$, and $(z^{j1}, \dots, z^{jm}) \in V_\epsilon(z^j, z^j, \delta)$ for $j = 1, \dots \ell$ be associated with K as in part (v) of Lemma 3.2 for $\epsilon > 0$ and $\delta > 0$ as given above. Thus, in particular,

$$\text{(7)} \qquad K \subset \bigcup_{j=1}^{\ell} \left(z^j + \tau_j \operatorname{int} \mathbb{B} \right).$$

By (5) and (7), there exists for each $k = 1, 2, \dots$ a $j_k \in \{1, 2, \dots, \ell\}$ such that

$$x^k \in z^{j_k} + \tau_{j_k} \operatorname{int} \mathbb{B}.$$

Hence, by part (v) of Lemma 3.2,

$$(z^{j_k 1}, \dots, z^{j_k m}) + \bar{\mu}\mathbb{B}^m \subset V_\epsilon(z^{j_k}, x^k, \delta) \subset D_\epsilon^m(x^k).$$

Since the x^{ki} $(i = 1, \dots, m)$ are uniformly distributed random variables supported on $D_\epsilon^m(x^k)$, it follows that

$$\operatorname{prob}\{x^{ki} \in z^{j_k i} + \bar{\mu}\mathbb{B}\} = \frac{\operatorname{vol}(\bar{\mu}\mathbb{B})}{\operatorname{vol}(\epsilon\mathbb{B})} = \left(\frac{\bar{\mu}}{\epsilon}\right)^n.$$

Hence,

$$\operatorname{prob}\{(x^{k1}, \dots, x^{km}) \in (z^{j_k 1}, \dots, z^{j_k m}) + \bar{\mu}\mathbb{B}^m\} = \left(\frac{\bar{\mu}}{\epsilon}\right)^{nm}.$$

Consequently, since event (4) has occurred, we have with probability 1 that there is an infinite subsequence $J \subset \mathbb{N}$ and a $\bar{j} \in \{1, \dots, \ell\}$ such that for all $k \in J$

$$\text{(8)} \qquad \gamma^{-1}t_k < \min\left\{\gamma, \frac{\epsilon}{3}\right\},$$

$$\text{(9)} \qquad x^k \in z^{\bar{j}} + \frac{\epsilon}{3}\mathbb{B}, \qquad \text{and}$$

$$(x^{k1}, \dots, x^{km}) \in (z^{\bar{j}1}, \dots, z^{\bar{j}m}) + \bar{\mu}\mathbb{B}^m \subset V_\epsilon(z^{\bar{j}}, z^{\bar{j}}, \delta) \subset D_\epsilon^m(z^{\bar{j}}),$$

which implies that for all $k \in J$

$$\text{(10)} \qquad \{x^{k0}, x^{k1}, \dots, x^{km}\} \subset z^{\bar{j}} + \epsilon\mathbb{B} \qquad \text{and}$$

$$\text{(11)} \qquad \eta \le \left\| g^k \right\| \le \operatorname{dist}\left(0 \,\middle|\, \hat{G}_k\right) \le \rho_\epsilon(z^{\bar{j}}) + \delta,$$

where $\hat{G}_k = \operatorname{conv}\{\nabla f(x^{k1}), \dots, \nabla f(x^{km})\}$.

By construction we have that

$$-\gamma^{-1}\beta t_k \left\| g^k \right\| \le f(x^k + \gamma^{-1}t_k d^k) - f(x^k) \quad \forall\, k \in J.$$

Theorem 3.3 yields for each $k \in J$ the existence of $\tilde{x}^k \in [x^k + \gamma^{-1}t_k d^k, x^k]$ and $v^k \in \bar{\partial}f(\tilde{x}^k)$ such that

$$f(x^k + \gamma^{-1}t_k d^k) - f(x^k) = \gamma^{-1}t_k \left\langle v^k, d^k \right\rangle.$$

Since $\|d^k\| = 1$, relations (8) and (9) imply that $\tilde{x}^k \in z^{\bar{j}} + \epsilon \mathbb{B}$ and so $v^k \in G_\epsilon(z^{\bar{j}})$ for all $k \in J$. In addition, the Lipschitz continuity hypothesis implies that $\|v^k\| \leq \kappa$ for all $k \in J$. Hence,

$$-\gamma^{-1}\beta t_k \|g^k\| \leq \gamma^{-1} t_k \langle v^k, d^k \rangle \quad \forall\, k \in J,$$

or, equivalently,

$$-\beta \|g^k\| \leq -\|g^k\| + \langle v^k - g^k, d^k \rangle \quad \forall\, k \in J,$$

which in turn implies that

(12) $$0 \leq (\beta - 1)\eta + \left\langle v^k - g^k, \frac{-g^k}{\|g^k\|} \right\rangle \quad \forall\, k \in J.$$

By combining (10) and (11) with Lemma 3.1, where $C = G_\epsilon(z^{\bar{j}})$, and then applying the bound (6), we find that

$$\left\langle v^k - g^k, \frac{-g^k}{\|g^k\|} \right\rangle \leq \left[\kappa\sqrt{\frac{2}{\eta}} + \sqrt{2\kappa + \delta} \right] \sqrt{\delta} < (1 - \beta)\frac{\eta}{2}.$$

Plugging this into (12) yields the contradiction

$$0 < \frac{1}{2}(\beta - 1)\eta \quad \forall\, k \in J.$$

Consequently, the event (4) cannot occur with positive probability, which proves that if $\hat{\mathcal{E}}$ occurs, then with probability 1 there is a subsequence $J \subset \mathbb{N}$ such that $\rho_\epsilon(x^k) \to 0$.

Finally, if $\bar{x}$ is a cluster point of the subsequence $\{x^k\}_J$ with $J \subset \mathbb{N}$, then

$$\mathrm{dist}\left(0 \,\big|\, \bar{\partial}_\epsilon f(x^k)\right) \to_J 0$$

since $0 \leq \mathrm{dist}\left(0 \,\big|\, \bar{\partial}_\epsilon f(x^k)\right) \leq \rho_\epsilon(x^k) \to_J 0$. Hence $0 \in \bar{\partial}_\epsilon f(\bar{x})$ for every cluster point $\bar{x}$ of the subsequence $\{x^k\}_J$ since the multifunction $\bar{\partial}_\epsilon f$ is closed. $\quad\square$

Theorem 3.4 tells us that when f has compact level sets then with probability 1 the GS algorithm generates a sequence of iterates having at least one cluster point that is Clarke ϵ-stationary. It is interesting to note that we have deduced this without having shown that the values $\|g^k\| = \mathrm{dist}\left(0 \,\big|\, G_k\right)$ converge to zero. Indeed, if this sequence of values did converge to zero, then every cluster point of the sequence $\{x^k\}$ would be a Clarke ϵ-stationary point. A convergence result of the type where every cluster point satisfies some kind of approximate stationarity condition is what is usually obtained in the smooth case. We now show that the GS algorithm also has this property under a smoothness hypothesis. In the same vein, if one assumes convexity, then a much stronger result is possible, and we cover this case as well in our next result. This is introduced to provide intuition into the behavior of the GS algorithm in the familiar settings of smoothness and convexity. By no means are we suggesting that the GS algorithm is a useful method when either smoothness or convexity is present.

COROLLARY 3.5. *Let the hypotheses of Theorem 3.4 hold.*

1. *If the function f is everywhere continuously differentiable, then either the GS algorithm terminates finitely at a point $\bar{x}$ for which $0 \in G_\epsilon(\bar{x}) = \bar{\partial}_\epsilon f(\bar{x})$ or the sequences $\{x^k\}$ and $\{\|g^k\|\}$ are such that $\|g^k\| \to 0$ and every cluster point $\bar{x}$ of the sequence $\{x^k\}$ satisfies $0 \in G_\epsilon(\bar{x}) = \bar{\partial}_\epsilon f(\bar{x})$.*

2. *If the function f is convex, then with probability 1 either the GS algorithm terminates finitely at a point $\bar{x}$ for which $0 \in \bar{\partial}_\epsilon f(\bar{x} + \epsilon\mathbb{B})$ or every cluster point $\bar{x}$ of the sequence $\{x^k\}$ satisfies*

(13) $$f(\bar{x}) \leq \min_{\mathbf{R}^n} f(x) + 2\kappa\epsilon,$$

where κ is any Lipschitz constant for f on $\mathcal{L}$.

Remark. Condition (13) is equivalent to the condition that $0 \in \partial_{2\kappa\epsilon} f(\bar{x})$, where $\partial_\epsilon f(x)$ denotes the ϵ-subdifferential from convex analysis:

$$\partial_\epsilon f(x) = \{z \mid f(y) \geq f(x) + \langle z, y - x \rangle - \epsilon \; \forall \, y \in \mathbf{R}^n \}.$$

Proof.

1. If the algorithm terminates finitely at iteration k_0, then it must do so in Step 2, in which case $0 \in G_\epsilon(\bar{x})$.

Next suppose that the algorithm does not terminate finitely. Further, let us suppose to the contrary that the sequence of values $\|g^k\|$ contains a subsequence $J \subset \mathbb{N}$ that is bounded away from zero:

$$0 < \eta := \inf_J \|g^k\|.$$

Since the sequence $\{f(x^k)\}$ is decreasing and the set $\mathcal{L}$ is compact, we may assume with no loss of generality that there is a point $\bar{x} \in \mathcal{L}$ such that $x^k \to_J \bar{x}$ and $f(x^k) \downarrow f(\bar{x})$. In addition, we have

$$f(x^{k+1}) - f(x^k) < -t_k \beta \|g^k\|,$$

and so $t_k \|g^k\| \to 0$. For the subsequence J this implies that $t_k \to 0$. Thus, we may assume again with no loss of generality that $t_k \downarrow_J 0$ with $t_k < 1$ for all $k \in J$. Hence, for all $k \in J$ there exists $z^k \in [x^k, x^k + \gamma^{-1} t_k d^k]$ such that

$$
\begin{aligned}
-\gamma^{-1} t_k \beta \|g^k\| &\leq f(x^k + \gamma^{-1} t_k d^k) - f(x^k) \\
&= \gamma^{-1} t_k \langle \nabla f(z^k), d^k \rangle \\
&= \gamma^{-1} t_k \langle \nabla f(x^k), d^k \rangle + \gamma t_k^{-1} \langle \nabla f(z^k) - \nabla f(x^k), d^k \rangle \\
&\leq \gamma^{-1} t_k \left[\sup_{g \in G_k} \langle g, d^k \rangle \right] + \gamma t_k^{-1} \|\nabla f(z^k) - \nabla f(x^k)\| \\
&= -\gamma^{-1} t_k \|g^k\| + \gamma t_k^{-1} \|\nabla f(z^k) - \nabla f(x^k)\|,
\end{aligned}
$$

where the first inequality follows since $t_k < 1$, the first equality is an application of the mean value theorem, and the third equality follows by construction since d^k solves the problem $\inf_{\|d\| \leq 1} \sup_{g \in G_k} \langle g, d \rangle$. Dividing through by $\gamma^{-1} t_k$ and rearranging yields the inequality

$$0 \leq (\beta - 1) \|g^k\| + \|\nabla f(z^k) - \nabla f(x^k)\| \leq (\beta - 1)\eta + \|\nabla f(z^k) - \nabla f(x^k)\|.$$

Taking the limit over J and using the continuity of ∇f we obtain the contradiction $0 \leq (\beta - 1)\eta < 0$. Hence $\|g^k\| \to 0$.

The final statement in part 1 of the corollary now follows from the inequality $0 \leq \text{dist}(0 \mid G_\epsilon(\bar{x})) \leq \|g^k\|$ and the continuity of the gradient.

2. Theorem 3.4 states that with probability 1 either the GS algorithm terminates finitely at a point $\bar{x}$ satisfying $0 \in G_\epsilon(\bar{x}) \subset \bar{\partial}_\epsilon f(\bar{x})$ or there is a cluster point $\hat{x}$ that is a Clarke ϵ-stationary point of f. Hence we need only focus on the case where we obtain the cluster point $\hat{x}$. Since the GS algorithm is a descent method we know that $f(\hat{x}) = f(\bar{x})$, where $\bar{x}$ is any cluster point of the sequence generated by the algorithm. Hence we need only show that (13) holds for $\hat{x}$.

Since $0 \in \bar{\partial}_\epsilon f(\hat{x})$, Carathéodory's theorem states that there exist pairs (z^1, w^1), $(z^2, w^2), \ldots, (z^{n+1}, w^{n+1}) \in \mathbf{R}^n \times \mathbf{R}^n$ and nonnegative scalars $0 \le \lambda_j \le 1$, $j = 1, \ldots, n+1$, such that $z^j \in \hat{x} + \epsilon \mathbb{B}$, $w^j \in \bar{\partial} f(z^j)$, $j = 1, \ldots, n+1$, $\sum_{j=1}^{n+1} \lambda_j = 1$, and $\sum_{j=1}^{n+1} \lambda_j w^j = 0$. The subdifferential inequality implies that for every $x \in \mathbf{R}^n$

$$(14) \qquad f(x) \ge f(z^j) + \langle w^j, x - z^j \rangle, \ \ j = 1, \ldots, n+1.$$

Let $w \in \partial f(\hat{x})$. Multiply each of the inequalities in (14) by its associated λ_j and sum up. Then

$$
\begin{aligned}
f(x) &\ge \sum_{j=1}^{n+1} \lambda_j f(z^j) + \sum_{j=1}^{n+1} \lambda_j \langle w^j, x - z^j \rangle \\
&\ge \sum_{j=1}^{n+1} \lambda_j (f(\hat{x}) + \langle w, z^j - \hat{x} \rangle) + \left\langle \sum_{j=1}^{n+1} \lambda_j w^j, x - \hat{x} \right\rangle + \sum_{j=1}^{n+1} \lambda_j \langle w^j, \hat{x} - z^j \rangle \\
&\ge f(\hat{x}) + \sum_{j=1}^{n+1} \lambda_j \langle w^j - w, \hat{x} - z^j \rangle \\
&\ge f(\hat{x}) - \sum_{j=1}^{n+1} \lambda_j (\|w^j\| + \|w\|) \|\hat{x} - z^j\| \\
&\ge f(\hat{x}) - \sum_{j=1}^{n+1} \lambda_j 2\kappa\epsilon \\
&= f(\hat{x}) - 2\kappa\epsilon,
\end{aligned}
$$

where $w \in \partial f(\hat{x})$ and we have used the fact that κ is a bound on any subgradient at a point in $\mathcal{L}$.   □

Our next convergence result is in the spirit of the *sequential unconstrained minimization technique* (SUMT) employed in [FM68]. Here we consider the behavior of the set of cluster points of the GS algorithm in fixed ϵ mode as ϵ is decreased to zero.

COROLLARY 3.6. *Let $\{\epsilon_j\}$ be a sequence of positive scalars decreasing to zero. For each $j = 1, 2, \ldots$ consider the GS algorithm with $\nu_0 = 0$ and $\epsilon = \epsilon_j$. Let C_j denote either the point of finite termination of the iterates or, alternatively, the set of all cluster points of the resulting infinite sequence of iterates. By Theorem 3.4, with probability 1 there exists a sequence $\{x^j\}$ with $x^j \in C_j$, $j = 1, 2, \ldots$, satisfying $0 \in \bar{\partial}_{\epsilon_j} f(x^j)$ for each $j = 1, 2, \ldots$. Then every cluster point of the sequence $\{x^j\}$ is a Clarke stationary point for f. Moreover, if the set $\mathcal{L}$ contains only one Clarke stationary point $\bar{x}$, in which case $\bar{x}$ is the strict global minimizer of f, then with probability 1 the entire sequence $\{x^j\}$ must converge to $\bar{x}$ and $\sup\{\|x - \bar{x}\| \mid x \in C_j\} \to 0$; that is, the sets C_j converge to the single point $\bar{x}$ in the Hausdorff sense.*

Proof. Let $\{\delta_j\}$ be any positive sequence of scalars decreasing to zero. Since $0 \in \bar{\partial}_{\epsilon_j} f(x^j)$ for each $j = 1, 2, \ldots$, Carathéodory's theorem tells us that for each

$j = 1, 2, \ldots$ there exists $\{x_1^j, \ldots, x_{n+1}^j\} \subset x^j + \epsilon_j \mathbb{B}$, $\{z_1^j, \ldots, z_{n+1}^j\} \subset \mathbf{R}^n$ with $z_s^j \in \bar{\partial} f(x_s^j)$, $s = 1, \ldots n + 1$, and $\lambda_1^j, \ldots, \lambda_{n+1}^j \in \mathbf{R}_+$ with $\sum_{s=1}^{n+1} \lambda_s^j = 1$ such that $\|\sum_{s=1}^{n+1} \lambda_s^j z_s^j\| \leq \delta_j$. Note that we need to choose the δ_j's strictly positive due to the closure operation in the definition of $\bar{\partial}_\epsilon f$. Let $\hat{x}$ be any cluster point of the sequence $\{x^j\}$ with associated subsequence $J \subset \mathbb{N}$ such that $x^j \to_J \hat{x}$. By compactness and the upper semicontinuity of $\bar{\partial} f$, we may assume with no loss in generality that there exist $\lambda_1, \ldots, \lambda_{n+1} \in \mathbf{R}_+$ with $\sum_{s=1}^{n+1} \lambda_s = 1$ and $z_s \in \bar{\partial} f(\hat{x})$ such that $\lambda_s^j \to_J \lambda_s$ and $z_s^j \to_J z_s$, $s = 1, \ldots, n + 1$. Then $\sum_{s=1}^{n+1} \lambda_s^j z_s^j \to_J \sum_{s=1}^{n+1} \lambda_s z_s \in \bar{\partial} f(\hat{x})$. But by construction $\|\sum_{s=1}^{n+1} \lambda_s^j z_s^j\| \leq \delta_j$ with $\delta_j \downarrow 0$. Hence it must be the case that $\sum_{s=1}^{n+1} \lambda_s z_s = 0$, which shows that $0 \in \bar{\partial} f(\hat{x})$.

Next assume that the set $\mathcal{L}$ contains only one Clarke stationary point $\bar{x}$. If the sequence $\{x^j\}$ does not converge to $\bar{x}$, then there is a subsequence that remains bounded away from $\bar{x}$. Since the GS algorithm only generates iterates in the compact set $\mathcal{L}$, this subsequence must have a cluster point in $\mathcal{L}$ that differs from $\bar{x}$. Since we have just shown that this cluster point must be a Clarke stationary point of f, we obtain the contradiction that establishes the result.

Finally suppose that $\bar{x}$ is a local minimizer of f. Since any local minimizer is a Clarke stationary point, we have that $\bar{x}$ is the unique local minimizer of f in the level set $\mathcal{L}$ and hence is the unique global minimizer of f in the level set $\mathcal{L}$ as well. Observe that since the GS algorithm is a descent algorithm, we have that $f(x) = f(x^j)$ for all $x \in C_j$ and all $j = 1, 2, \ldots$. We have just shown that $x^j \to \bar{x}$; hence $f(x^j) \to f(\bar{x})$. Now if the sequence of values $\sup \{\|x - \bar{x}\| \mid x \in C_j\}$ does not converge to zero, then there must be a sequence $\{\hat{x}^j\}$ with $\hat{x}^j \in C_j$ for each $j = 1, 2, \ldots$ such that the sequence is bounded away from $\bar{x}$. Due to compactness there is a subsequence $J \subset \mathbb{N}$ such that $\{\hat{x}^j\}$ converges to some $\hat{x} \in \mathcal{L}$. But then $f(\bar{x}) = \lim_J f(x^j) = \lim_J f(\hat{x}^j) = f(\hat{x})$. Therefore, $\hat{x}$ must also be a global minimizer of f on $\mathcal{L}$. Since this global minimizer is unique, we arrive at the contradiction $\hat{x} = \bar{x}$, which proves the result. $\square$

COROLLARY 3.7. *Suppose that the set $\mathcal{L}$ contains a unique Clarke stationary point $\bar{x}$ of f, in which case $\bar{x}$ is the unique global minimizer of f. Then for every $\delta > 0$ there exists an $\bar{\epsilon} > 0$ such that if the GS algorithm is initiated with $\epsilon \in (0, \bar{\epsilon})$, then with probability 1 either the algorithm terminates finitely at a point within a distance δ of $\bar{x}$ or, alternatively, every cluster point of the resulting infinite sequence of iterates is within a distance δ of $\bar{x}$.*

Proof. Suppose the result is false. Then with positive probability there exists $\delta > 0$ and a sequence $\epsilon_j \downarrow 0$ such that the set of cluster points C_j (or the finite termination point) of the sequence of iterates generated by the GS algorithm with $\epsilon = \epsilon_j$ satisfy $\sup \{\|x - \bar{x}\| \mid x \in C_j\} > \delta$. But this contradicts the final statement of Corollary 3.6 whereby the result is established. $\square$

The convergence results stated above describe the behavior of the GS algorithm in fixed ϵ mode. We now give a final convergence result for the algorithm in the case where ϵ_k and ν_k are allowed to decrease.

THEOREM 3.8. *Let $\{x^k\}$ be a sequence generated by the GS algorithm with $\nu_0 > 0$, $\epsilon_0 > 0$, $\mu \in (0, 1)$, and $\theta \in (0, 1)$. With probability 1 the sequence $\{x^k\}$ is infinite. Moreover, if the sequence $\{x^k\}$ converges to some point $\bar{x}$, then, with probability 1, $\nu_k \downarrow 0$ and $\bar{x}$ is a Clarke stationary point for f.*

Proof. We restrict our discussion to instances of the GS algorithm in $\mathcal{E}$. The algorithm terminates finitely only if it terminates in either Step 1 or 2. As previously observed, finite termination in Step 1 has zero probability. Finite termination occurs

in Step 2 if $\nu_k = \left\|g^k\right\| = 0$. But by construction, $0 < \nu_k$ for all k if $0 < \nu_0$. Hence the algorithm cannot terminate in Step 2. Therefore, with probability 1 the sequence $\{x^k\}$ is infinite.

Let us first observe that if $\nu_k \downarrow 0$, then every cluster point is a Clarke stationary point due to the upper semicontinuity of $\bar{\partial}f$ and the relation

$$\mathrm{dist}\left(0 \mid \bar{\partial}_{\epsilon_k}f(x^k)\right) \leq \mathrm{dist}\left(0 \mid G_{\epsilon_k}(x^k)\right) \leq \mathrm{dist}\left(0 \mid G_k\right) = \left\|g^k\right\| \leq \nu_k.$$

In this case it follows that the entire sequence must converge to $\bar{x}$ since $\bar{x}$ is the unique Clarke stationary point.

Hence, we may as well assume that

$$\bar{\nu} = \inf_k \nu_k > 0.$$

If $\bar{\nu} > 0$, then it must be the case that ν_k and ϵ_k were updated only finitely many times. That is, there is an index k_0 and an $\bar{\epsilon} > 0$ such that

$$\nu_k = \bar{\nu} \quad \text{and} \quad \epsilon_k = \bar{\epsilon} \quad \forall\, k \geq k_0.$$

This places us in the context of Theorem 3.4 where finite termination almost surely does not occur since $\epsilon_k = \bar{\epsilon}$ and $\left\|g^k\right\| > \nu_k = \bar{\nu}$ for all k sufficiently large. Hence the cluster point $\bar{x}$ must be a Clarke $\bar{\epsilon}$-stationary point of f. By part (i) of Lemma 3.2, there is a $\tau > 0$ and a nonempty open set $\bar{V}$ such that the set $V_\epsilon(\bar{x}, x, \bar{\nu}/2)$ contains $\bar{V}$ whenever $\|x - \bar{x}\| \leq \tau$. Now since we assume event $\mathcal{E}$, the sequence $\{(u^{k1}, \ldots, u^{km})\}$ hits every open subset of the unit ball $\mathbb{B}^m$ infinitely often, or, equivalently, the sequence $\{(\bar{\epsilon}u^{k1}, \ldots, \bar{\epsilon}u^{km})\}$ hits every open subset of $\bar{\epsilon}\mathbb{B}^m$ infinitely often. As $x^k \to \bar{x}$, we have that the sequence $\{(x^{k1}, \ldots, x^{km})\}$ hits $\bar{V}$ infinitely often. But then it must be the case that $\left\|g^k\right\| \leq \bar{\nu}/2$ infinitely often since $\rho_\epsilon(\bar{x}) = 0$. This is the contradiction that proves the result.   □

We end this section with a summary of some open questions.

1. Theorem 3.4 indicates that the algorithm may not terminate. Under what conditions can one guarantee that the GS algorithm terminates finitely?
2. In Theorem 3.4, we show that if the GS algorithm does not terminate finitely, then there exists a subsequence $J \subset \mathbb{N}$ such that $\rho_\epsilon(x^k) \to_J 0$. But we cannot show that the corresponding subsequence g^k converges to 0. Can one show that $\left\|g^k\right\| \to_J 0$? Or, is there a counterexample? We believe that a counterexample should exist.
3. We have successfully applied the GS algorithm in many cases where the function f is not Lipschitz continuous. Is there an analogue of Theorem 3.4 in the non-Lipschitzian case?
4. Is it possible to remove the uniqueness hypothesis in Corollary 3.7?
5. In Theorem 3.8, can one show that all cluster points of the sequence are Clarke stationary points?

4. Numerical results. We have had substantial experience using the GS algorithm to solve a wide variety of nonsmooth, nonconvex minimization problems that arise in practice. In this section we describe some of these problems and present some of the numerical results. As far as we are aware, none of the problems we describe here has been solved previously by any method.

We begin by describing our choices for the parameters defining the GS algorithm as well as changes that must be made to implement it in finite precision. We have

attempted to minimize the discrepancies between the theoretical and implemented versions of the method, but some are unavoidable. The algorithm was implemented in MATLAB, which uses IEEE double precision arithmetic (thus accuracy is limited to about 16 decimal digits). For the solution of the convex quadratic program required in Step 2, we tried several QP solvers; we found MOSEK [Mos03] to be the best of these in terms of reliability, efficiency, and accuracy.

Sample size. Bearing in mind the requirement that the sample size m must be greater than n, the number of variables, we always set $m = 2n$.

Sampling parameters. We used $\epsilon_0 = \mu = 0.1$, thus initializing the sampling radius to 0.1 and reducing it by factors of 0.1. Naturally, appropriate choices depend on problem scaling; these values worked well for our problems.

Optimality tolerance parameters. We used $\nu_0 = 10^{-6}$ and $\theta = 1$, thus fixing the optimality tolerance to 10^{-6} throughout. (We experimented with $\theta < 1$, so that a coarser sampling radius is associated with a coarser optimality tolerance; this reduced the iteration counts for easier problems but led to difficulty on harder ones.)

Line search parameters. We set the backtracking reduction factor γ to the standard choice of 0.5, but we set the Armijo parameter β to the decidedly nonstandard choice of 0. The theoretical analysis requires $\beta > 0$, but in practice, on difficult problems, even a modest "sufficient decrease" test can cause the algorithm to fail prematurely, and we never encountered any convergence difficulty that could be attributed to the choice $\beta = 0$. Setting β to a very small number such as 10^{-16} is, for all practical purposes, equivalent to setting it to zero.

Maximum number of iterations and line search failure. We limited the number of iterations for each sampling radius to 100; once the limit of 100 is reached, the sampling radius is reduced just as if the condition $\|g^k\| \leq \nu_k = 10^{-6}$ in Step 2 were satisfied, so the line search is skipped, and sampling continues with the smaller sampling radius. The smallest sampling radius allowed was 10^{-6}; instead of reducing it to 10^{-7}, the algorithm terminates. Thus, the total number of iterations is at most 600. Also, the line search may fail to find a lower function value (either because a limit on the number of backtracking steps, namely, 50, is exceeded, or because the computed direction of search is actually not a descent direction). Line search failure is quite common for the more difficult problems and generally indicates that, roughly speaking, the maximum accuracy has been achieved for the current sampling radius. When line search failure occurs, we reduce the sampling radius and continue, just as if the condition $\|g^k\| \leq \nu_k = 10^{-6}$ were satisfied or the limit of 100 iterations were reached.

Skipping the differentiability check. We do not attempt to check whether the iterates lie in the set D where f is differentiable, in either Step 1 or Step 4. This is simply impossible in finite precision and in any case would make life very difficult for the user who provides function and gradient values. The user need not be concerned about returning special values if the gradient is not defined at a point; typically, this happens because a "tie" takes place in the evaluation of the function, and the user may simply break the tie arbitrarily. The justification for this is that, for all practical purposes, in finite precision the set D is never encountered except in contrived, trivial cases.

Ensuring boundedness. In practice it is advisable to terminate the algorithm if an a priori bound on the norm of the iterates x^k is exceeded; we set this bound to 1000, but it was not activated in the runs described here.

All parameters and limits described above are easily changed by users of our

MATLAB implementation.

We now present a selection of numerical results for the GS algorithm, concentrating on problems that have not, to our knowledge, been solved previously. In the tables below, each line corresponds to running the GS algorithm on one problem (one instance of f). Because of the stochastic nature of the algorithm, we ran it 10 times, either from one given starting point, when specified, or from 10 different random starting points, when so indicated; the results shown are for the run achieving the lowest value of f. For each problem class, we display the results for a range of problems, ranging from easiest to hardest, and collected in a single table. Each line of every table displays the final value of f, the total number of iterations for all six sampling radii (the total number of times Step 2 was executed), and an approximate "optimality certificate." The last deserves a detailed explanation. An approximate optimality certificate consists of two numbers; the first is an "optimality residual norm" $\|g^k\|$, and the second is the value of the sampling radius ϵ_k for which the first value $\|g^k\|$ was achieved. These quantities together provide an estimate of nearness to Clarke stationarity. Instead of simply displaying the final optimality certificate, we show the certificate for the smallest sampling radius ϵ_k for which the test $\|g^k\| \leq \nu_k = 10^{-6}$ was satisfied, or, if it was satisfied for no ϵ_k, simply the final values.

We note that for the problems described in sections 4.1 through 4.4 we think that the local minimizers approximated by the GS algorithm are in fact global minimizers, based on the failure to find other locally minimal optimal values when initializing the algorithm at other starting points.[2] However, we discuss the difficulty of finding global minimizers in section 4.5.

We also remark that, for comparison purposes, we have attempted to solve the same problems by other methods, particularly the Bundle Trust (BT) FORTRAN code of [SZ92]. It is faster than our code but, in our experience, generally provides less accurate results and is unable to solve any of the harder problems described below. We also experimented with a variety of "direct search" methods which are not intended for nonsmooth problems but are so robust that they are worth trying anyway. Of these, the most successful was the well-known Nelder–Mead method, but it was only able to solve the easier problems with very small size n. An important observation is that a user of the BT code or Nelder–Mead method generally has no way of knowing how good a computed approximation might be in the absence of any kind of local optimality certificate.

The data matrices for the problems discussed in sections 4.2 and 4.5 are available on the Web, as are the computed solutions that we obtained for all the problems.[3]

4.1. Chebyshev approximation by exponential sums. Our first example is a classical one: Chebyshev approximation. The function to be minimized is

$$f(x) = \sup_{s \in [\ell, u]} |h(s, x)|,$$

where $[\ell, u]$ is any real interval and $h : \mathbf{R} \times \mathbf{R}^n \to \mathbf{R}$ is any smooth function. To evaluate $f(x)$ for a given $x \in \mathbf{R}^n$, we evaluate $h(\cdot, x)$ on a one-dimensional grid, find the maximum (in absolute value), and use this to initialize a one-dimensional local maximization method, based on successive cubic interpolation using the derivative of h with respect to s, to accurately locate a maximizer, say $\bar{s}$. The finer the grid is, the

---

[2] A possible exception is the problem defined by $N = 8$ in section 4.2.
[3] http://www.cs.nyu.edu/overton/papers/gradsamp/probs/.

TABLE 1
Results for exponential Chebyshev approximation, starting from $x = 0$.

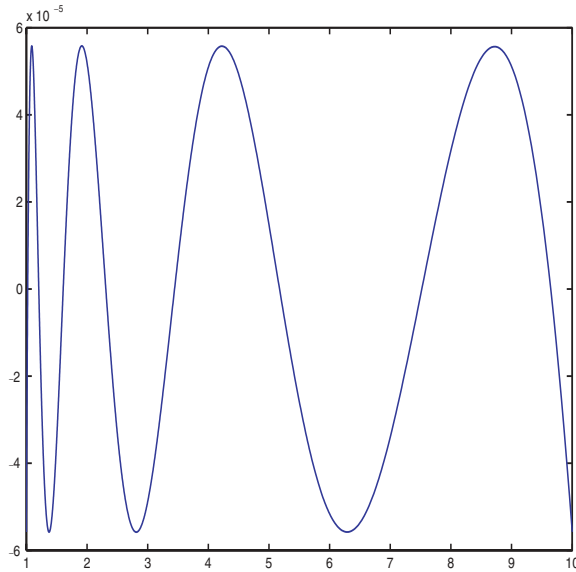| n	f	Opt cert	Iters
2	8.55641e-002	(9.0e-011, 1.0e-004)	42
4	8.75226e-003	(8.9e-009, 1.0e-006)	63
6	7.14507e-004	(6.5e-007, 1.0e-004)	166
8	5.58100e-005	(2.2e-005, 1.0e-006)	282

FIG. 1. *The error function for the sum of exponentials approximation to $1/s$ on $[1, 10]$ for $n = 8$, with nine alternation points.*

more likely one is to obtain the global maximizer. The function f is differentiable if the maximizer is unique, with gradient

$$\nabla f(x) = \text{sign}(h(\bar{s}, x))\nabla h_x(\bar{s}, x).$$

We use

$$h(s, x) = \frac{1}{s} - \sum_{j=1}^{n/2} x_{2j-1} \exp(-x_{2j}s),$$

where n is even and $\ell > 0$. Thus the problem is to approximate the function $1/s$ on a positive interval by a sum of decaying exponentials. We chose $[\ell, u] = [1, 10]$ with 2000 grid points, equally spaced in the target function value $1/s$.

Table 1 shows the results obtained by the GS algorithm (see above for interpretation of the optimality certificate "opt cert"). We may safely conjecture on the basis of these results that the optimal value decays exponentially with n. The accuracy achieved is of course limited by the conditioning of the problem and the finite precision being used: accurate solutions were not obtainable for $n > 8$.

Figure 1 shows the error function $h(s, x)$ as a function of s for the optimal parameter vector x found for $n = 8$. Notice the alternation in the error function; the

TABLE 2
Results for minimizing eigenvalue product, using random starting points.

n	N	k	f	Mult	Opt cert	Iters
1	2	1	1.00000e+000	1	(1.3e-008, 1.0e-006)	10
6	4	2	7.46286e-001	2	(7.3e-006, 1.0e-006)	68
15	6	3	6.33477e-001	2	(5.8e-006, 1.0e-006)	150
28	8	4	5.58820e-001	4	(1.0e-001, 1.0e-006)	600
45	10	5	2.17193e-001	3	(1.7e-005, 1.0e-006)	277
66	12	6	1.22226e-001	4	(9.7e-003, 1.0e-006)	432
91	14	7	8.01010e-002	5	(4.5e-006, 1.0e-006)	309
120	16	8	5.57912e-002	6	(2.7e-003, 1.0e-006)	595

maximum error is achieved at nine places on the interval (the leftmost one being essentially invisible). Work of Rice in the 1950s [Mei67] showed that if an optimal approximation exists, such alternation must take place, with the number of alternation points equal to one plus the number of parameters, but computation of such an error function was impossible at the time. A picture like Figure 1 may well have appeared in the more recent literature (perhaps computed by semi-infinite programming), but we have not seen one.

We remark that the optimal approximations seem to be unique up to the obvious permutation of pairs of parameters with one another; the ordering of pairs (x_{2j-1}, x_{2j}) is arbitrary.

4.2. Minimization of eigenvalue products. Our second example is the following problem: minimize the product of the largest k eigenvalues of a Hadamard (componentwise) matrix product $A \circ X$, where A is a fixed positive semidefinite symmetric matrix and X is a variable symmetric matrix constrained to have ones on its diagonal and to be positive semidefinite. Since the latter constraint is convex, we could impose it via projection, but for simplicity we handle it by an exact penalty function. Thus the function to be minimized is

$$f(x) = \prod_{j=1}^{k} \lambda_j(A \circ X) - \rho \min(0, \lambda_N(X)),$$

where λ_j means jth largest eigenvalue and the $N \times N$ symmetric matrix X has ones on its diagonal and $n = N(N-1)/2$ variables from the vector x in its off-diagonal positions. We set $\rho = 100$. The function f is differentiable at a vector x corresponding to a matrix X if X is positive definite and $\lambda_k(A \circ X) > \lambda_{k+1}(A \circ X)$. The gradient of f at such points is easily computed using the chain rule and the fact that the derivative of a simple eigenvalue λ_j in matrix space is the outer product qq^T defined by its corresponding normalized eigenvector q. As explained earlier, the user coding the gradient need not be concerned about ties, whether these are ties for the choice of kth eigenvalue of $A \circ X$, ties for the ordering of its eigenvalues λ_j for $j < k$, or for the boundary case $\lambda_N(X) = 0$.

Table 2 shows results for various instances of this problem: The matrices A are the leading $N \times N$ submatrices of a specific 63×63 covariance data matrix arising in an environmental application [AL04]. In each case we set k, the number of eigenvalues in the product, to $N/2$. For each minimizer approximated for $N > 2$, the matrix $A \circ X$ has a multiple interior eigenvalue including λ_k; its multiplicity is shown in the table. In addition, for $N > 4$, the minimizer X has a multiple zero eigenvalue.

The results in Table 2 demonstrate that use of the GS algorithm is by no means

restricted to very small n. Each iteration of the algorithm requires the solution of a quadratic program in m variables, a cost that is a small degree polynomial in n since $m = 2n$. However, solving this problem for $N > 20$ ($n > 200$) would take an unreasonable amount of computer time at present.

4.3. Spectral and pseudospectral minimization. We now enter the realm of nonsymmetric real matrices, whose eigenvalues may be complex and are non-Lipschitz at some points in matrix space. We are interested in a function known as the pseudospectral abscissa of a matrix, $\alpha_\delta(X)$, defined, for any given $\delta \geq 0$, as the maximum of the real parts of the δ-pseudospectrum of X, that is, the set of all z in the complex plane such that z is an eigenvalue of some complex matrix within a distance δ of X [Tre97]. Here, distance is measured in the operator 2-norm. Pseudospectra, and more specifically the pseudospectral abscissa, arise naturally in the study of robust stability of dynamical systems. When $\delta = 0$ the pseudospectral abscissa reduces to the spectral abscissa (the maximum of the real parts of the eigenvalues of the given matrix X). An algorithm for computing α_δ was given by the authors in [BLO03b]. As is the case with so many of the applications we have encountered, computing the function value is quite complicated, but once it is computed, the gradient is easy to obtain where defined, in this case requiring only the computation of singular vectors corresponding to a certain least singular value. As usual, the user coding the gradient need not be concerned with ties for the minimum value.

We consider a simple parameterized matrix,

$$(15) \qquad X(x) = \begin{bmatrix} -x_1 & 1 & 0 & \cdot & \cdot & 0 \\ x_1 & 0 & 1 & 0 & \cdot & 0 \\ x_2 & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ x_n & 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix},$$

where n, the number of parameters, is one less than the order of the matrix, say N. Our optimization problem is to minimize

$$(16) \qquad f(x) = \alpha_\delta(X(x))$$

over the parameter vector x. The authors showed in [BLO01] that, in the case $\delta = 0$, the global minimizer of f is 0. It is easy to verify in this case that f is not Lipschitz at 0; in fact, f grows proportionally with $|x_n|^{1/N}$. The authors have also shown [BLO03a] that, for fixed small positive δ, f is Lipschitz near 0, but it is not known whether this is true on the whole parameter space.

Figure 2 shows the optimal values of (16) found by the GS algorithm for various values of δ and N. We used $x = 0$ as the starting point since this is the optimal solution for $\delta = 0$. The figure suggests a conjecture: that the optimal value is proportional to $\delta^{2/(N+1)}$ for small δ. The irregularity at the top right of the plot is not numerical error, but a reminder that the phenomenon we are studying is nonlinear. We verified that the function (16) is indeed nonsmooth at all the minimizers approximated by the GS algorithm.

Table 3 shows more detailed results for $N = 5$. Note particularly the final line in the table, which shows the case $\delta = 0$ (minimizing the pure spectral abscissa). Since the solution is $x = 0$, we initialized the runs randomly in this case. Because the exact optimal value of f is 0, the computed value of f necessarily has no correct
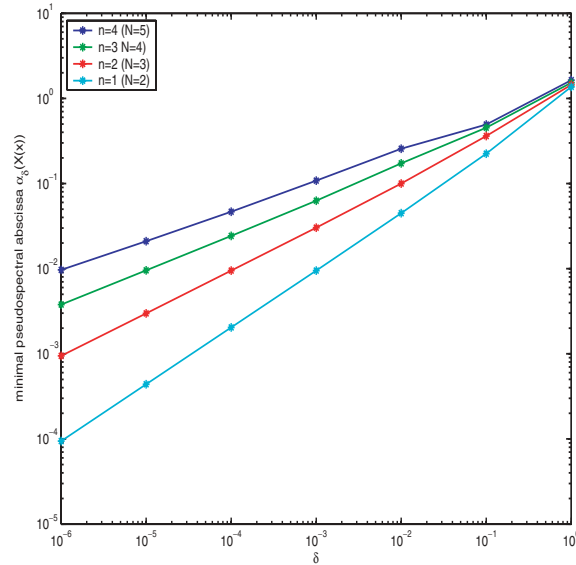
FIG. 2. *The minimal pseudospectral abscissa $\alpha_\delta(X(x))$ plotted as a function of δ, for various n.*

TABLE 3
Results for minimizing pseudospectral abscissa $\alpha_\delta(X(x))$ for $n = 4$ ($N = 5$), starting from $x = 0$ (except pure spectral abscissa case $\delta = 0$, started randomly).

δ	f	Opt cert	Iters
1	1.63547e+000	(8.4e-007, 1.0e-006)	81
1.0e-001	4.92831e-001	(4.5e-006, 1.0e-006)	105
1.0e-002	2.56467e-001	(7.4e-009, 1.0e-002)	112
1.0e-003	1.08221e-001	(7.6e-008, 1.0e-003)	163
1.0e-004	4.66477e-002	(3.2e-010, 1.0e-005)	236
1.0e-005	2.10125e-002	(3.0e-007, 1.0e-006)	322
1.0e-006	9.68237e-003	(6.3e-007, 1.0e-006)	403
0	4.03358e-003	(3.0e-007, 1.0e-006)	157

digits. However, its order of magnitude is about as good as can be expected using a precision of 16 decimal digits, because the exact spectral abscissa of $X(x)$ has order of magnitude $(10^{-15})^{1/5} = 10^{-3}$ for $\|x\| = 10^{-15}$, the approximate rounding level. This experiment indicates that the GS algorithm has no inherent difficulty with minimizing functions that are non-Lipschitz at their minimizers.

4.4. Maximization of distance to instability. A stable matrix is one with all its eigenvalues in the open left half-plane. The matrix $X(x)$ defined in (15) is not stable for any x, but the shifted matrix $X(x) - sI$ is stable for all $s > 0$ and sufficiently small $\|x\|$. Given a matrix X, its distance to instability, denoted $d_{\text{inst}}(X)$, is the least value δ such that some complex matrix Y within a distance δ of X is not stable. The distance to instability is a well-studied function [Bye88], especially in robust control, where it is known as the complex stability radius (or, more generally, as the inverse of the H_∞-norm of a transfer function) [BB90]. The relationship between α_δ and d_{inst} is summarized by

$$\alpha_\delta(X) = 0 \quad \text{for} \quad \delta = d_{\text{inst}}(X)$$
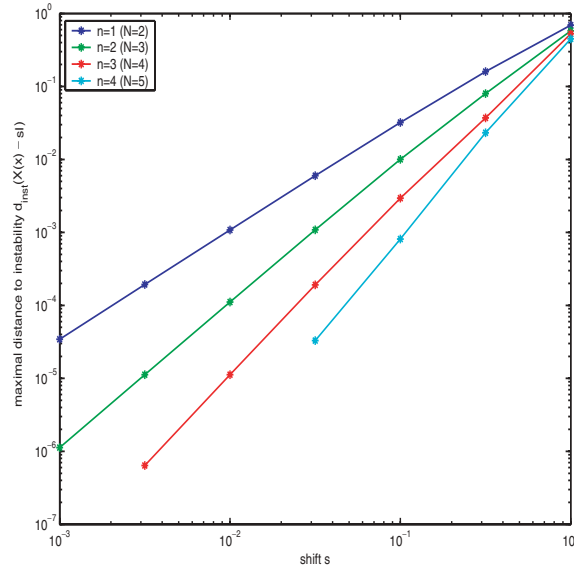
FIG. 3. *The maximal distance to instability $d_{\mathrm{inst}}(X(x) - sI)$ plotted as a function of the shift s, for various n.*

TABLE 4
Results for maximizing distance to instability $-f(x) = d_{\mathrm{inst}}(X(x) - sI)$ for $n = 4$ ($N = 5$), starting from $x = 0$.

s	f	Opt cert	Iters
1	-4.49450e-001	(1.4e-006, 1.0e-006)	55
3.16228e-001	-2.31760e-002	(1.5e-005, 1.0e-006)	71
1.00000e-001	-8.12170e-004	(8.5e-007, 1.0e-002)	110
3.16228e-002	-3.28692e-005	(1.8e-006, 1.0e-006)	141

for all stable X. By definition, $d_{\mathrm{inst}}(X) = 0$ if X is not stable.

We now consider the problem of maximizing $d_{\mathrm{inst}}(X(x) - sI)$ (equivalently, minimizing $f(x) = -d_{\mathrm{inst}}(X(x) - sI)$) over the parameter vector x, given $s > 0$. This is a difficult problem for small s because the set of x for which $f(x) < 0$ shrinks to 0 as $s \to 0$. We use the starting point $x = 0$ since $f(0) < 0$ for all $s > 0$. Figure 3 shows the optimal values found by the GS algorithm for various s and N. The missing data points in the table were suppressed because the computed values were too inaccurate to be meaningful. The figure suggests another conjecture, related to the one in the previous subsection: that the optimal value is proportional to $s^{(N+1)/2}$.

Table 4 gives details for $N = 5$.

4.5. Static output feedback and low-order controller design. Suppose the following are given: an $N \times N$ matrix A associated with a dynamical system $\dot{\xi} = A\xi$, together with an $N \times m$ matrix B (defining controllability of the system) and a $p \times N$ matrix C (defining observability of the system), with $m < N$ and $p < N$. Then, given an integer $k < N$, the *order k controller* design problem is to find X_1, X_2, X_3, and X_4, respectively with dimensions $m \times p$, $m \times k$, $k \times p$, and $k \times k$, such
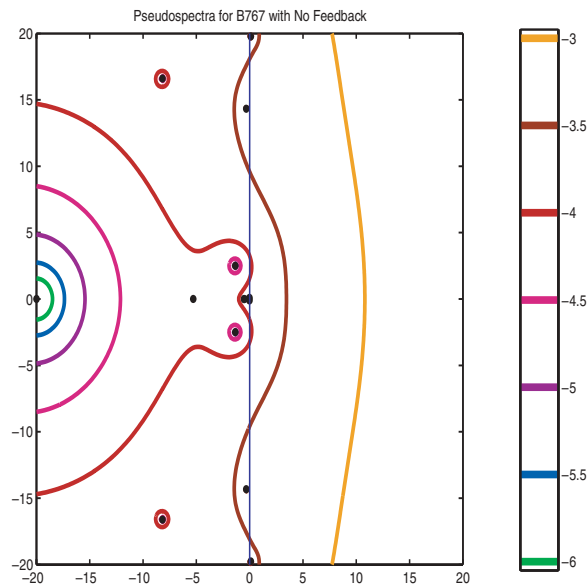
FIG. 4. *Eigenvalues and pseudospectra of B767 model at flutter condition with no controller.*

that the matrix describing the controlled system, namely,

$$(17) \qquad \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} X_1 & X_2 \\ X_3 & X_4 \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix},$$

satisfies desired objectives. We confine our attention to optimizing the following functions of this matrix: asymptotic stability, as measured by α_0, and robust stability, as measured by d_{inst}, respectively defined in the previous two subsections. When $k = 0$, the controlled system reduces to static (or memoryless) output feedback (SOF), the most basic control model possible, with just mp free variables. Clearly, one may think of the order k controller design problem as an SOF problem with $(m+k)(p+k)$ variables instead of mp, redefining A, B, and C as the larger block matrices in (17).

When k, m, and p are sufficiently large, it is known that stabilization is generically possible and there are various well-known techniques for finding such solutions [Won85, Wil97]. However, for $k, m, p \ll N$, how to efficiently find stabilizing X_1, X_2, X_3, X_4 (or show that this is not possible) is a long-standing open problem in control [BGL95]. The title of this subsection reflects the fact that we are interested only in small k.

We focus on a specific, difficult example, which arises from a model of a Boeing 767 at a flutter condition [Dav90]. The matrix A in this case describes a linearized model when flutter has started to occur; in other words, the plane is flying so fast that the aerodynamic and structural forces are interacting to generate an instability in the system. The matrix A has size $N = 55$, but the controllability and observability matrices B and C have only $m = 2$ columns and $p = 2$ rows, respectively. Figure 4 shows the eigenvalues and δ-pseudospectra boundaries of A as points (solid dots) and curves in the complex plane. (Recall from section 4.3 that the δ-pseudospectrum of A is the set of complex numbers z such that z is an eigenvalue of a complex matrix within a distance δ of A.) The legend on the right shows the values of δ using a log 10 scale. Note the complex conjugate pair of unstable eigenvalues near the top
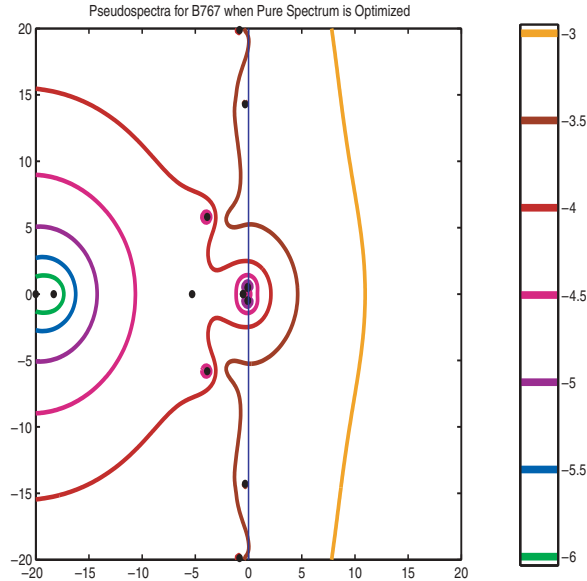
FIG. 5. *Eigenvalues and pseudospectra of B767 model when spectral abscissa is minimized for SOF (order 0 controller).*

and bottom of the figure. This plot and subsequent ones were drawn using Wright's software EigTool [Wri02].

We first investigated the case $k = 0$ (SOF), applying the GS algorithm to minimize the spectral abscissa α_0 of the matrix (17) over the four free parameters (the entries in X_1). Hundreds of runs from randomly chosen starting points repeatedly found the same unstable local minimizers, with $\alpha_0 > 0$. Eventually, however, the GS algorithm found a stable local minimizer, with $\alpha_0 = -7.79 \times 10^{-2}$ and an optimality certificate $(2.8 \times 10^{-15}, 10^{-5})$. The reason it was so difficult to find this minimizer is that the data is very badly scaled. Once it became evident what scaling to use for the starting point, the GS algorithm had no difficulty repeatedly finding this minimizer. Figure 5 shows the eigenvalues and pseudospectra of the stabilized matrix. Although all the eigenvalues are now (barely) to the left of the imaginary axis, even the 10^{-6}-pseudospectrum extends into the right half-plane. Thus, the matrix is not robustly stable: Tiny perturbations to it can generate instability.

We then used this stabilizing minimizer, as well as randomly generated small relative perturbations of it, as starting points for maximizing d_{inst} over the same four variables. The GS algorithm found a local optimizer with $d_{\text{inst}} = 7.91 \times 10^{-5}$ and optimality certificate $(9.2 \times 10^{-7}, 10^{-6})$, whose eigenvalues and pseudospectra are shown in Figure 6. Notice that the 10^{-5}-pseudospectrum now lies in the left half-plane, but that the 10^{-4}-pseudospectrum still extends into the right half-plane.

We now turn to order 1 and order 2 controllers ($k = 1$ and $k = 2$, respectively). We used the local optimizer for $k = 0$ as a starting point, as well as randomly generated small relative perturbations, to maximize the same d_{inst} objective over the 9-variable parametrization for an order 1 controller and the 16-variable parametrization for an order 2 controller. For $k = 1$, the GS algorithm found a local optimizer with $d_{\text{inst}} = 9.98 \times 10^{-5}$ and with optimality certificate $(7.9 \times 10^{-7}, 10^{-4})$, whose eigenvalues and pseudospectra are shown in Figure 7. For $k = 2$, the GS algorithm found a local
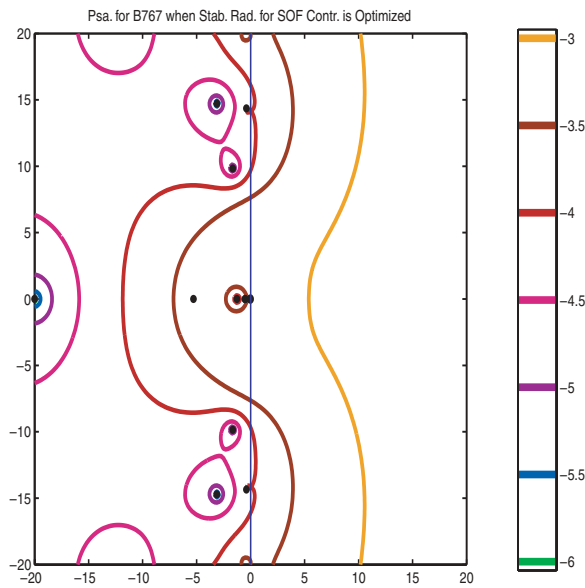
FIG. 6. *Eigenvalues and pseudospectra of B767 model when distance to instability is maximized for SOF model (order 0 controller).*
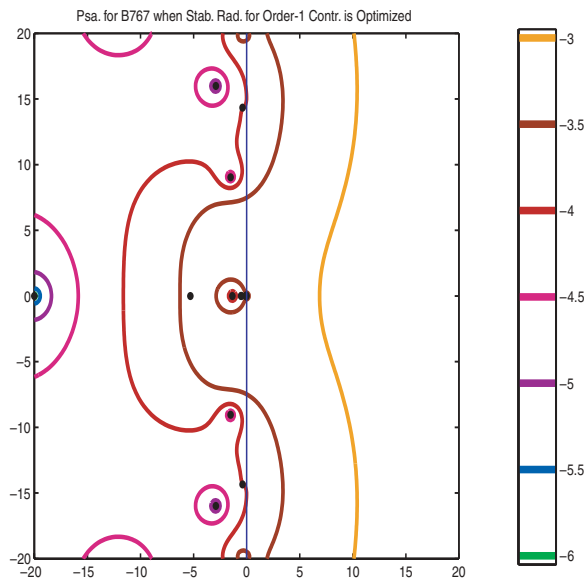

FIG. 7. *Eigenvalues and pseudospectra of B767 model when distance to instability is maximized for order 1 controller.*

optimizer with $d_{\text{inst}} = 1.02 \times 10^{-4}$ and with optimality certificate $(7.3 \times 10^{-6}, 10^{-6})$, whose eigenvalues and pseudospectra are shown in Figure 8. For $k = 1$, the 10^{-4}-pseudospectrum extends just slightly into the right half-plane, while for $k = 2$, it is barely to the left of the imaginary axis, indicating that perturbations of magnitude 10^{-4} or less cannot destabilize the matrix.
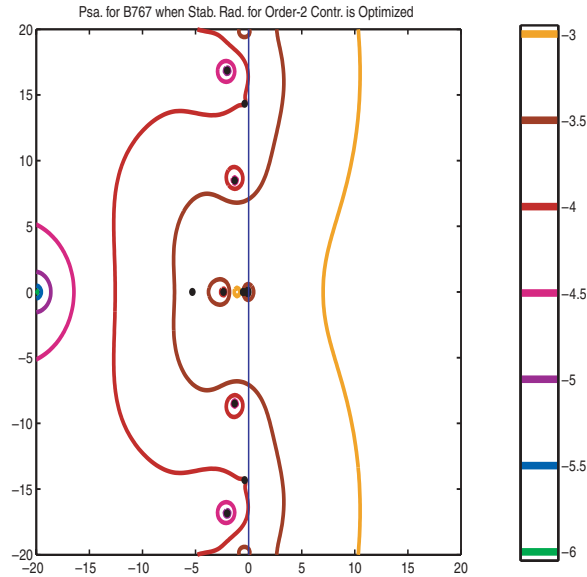
FIG. 8. *Eigenvalues and pseudospectra of B767 model when distance to instability is maximized for order 2 controller.*

As far as we are aware, no such low-order stabilizing controllers were known for the Boeing 767 model before we conducted this work. The optimality certificates that we obtained give us confidence that the optimizers we approximated are indeed local optimizers. However, our initial difficulty in finding even one stabilizing local minimizer of α_0 in the case $k = 0$ illustrates how difficult it is to find global minimizers, and we certainly cannot conclude that the local optimizers we found are global optimizers.

5. Concluding remarks. We have presented a new algorithm for nonsmooth, nonconvex optimization, proved its convergence to Clarke stationary points under strong assumptions, raised questions about other possible convergence results under weaker assumptions, extensively tested the algorithm, presented solutions of quite a number of interesting optimization problems that have not been solved previously, and shown how approximate first-order optimality certificates may be used to give some confidence that the solutions found are meaningful. We make a few final remarks.

All of the functions that we have minimized by the GS algorithm are subdifferentially regular (in the sense of Clarke; see section 1) at the minimizers that we found. We view regularity as a fundamental property that is crucial for the understanding of an optimization problem when smoothness and convexity, both of which are essentially special cases, are lacking. It is regularity that combines with Clarke stationarity to give a genuine first-order optimality condition: that the ordinary directional derivative is nonnegative in all directions. We have been able to show, although we do not give details here, that Chebyshev approximation error, eigenvalue products for symmetric matrices, and minus the distance to instability (on the stable domain) are globally regular functions using results like [RW98, Theorem 10.31] and [Lew99, Corollary 4]. The case of the pseudospectral abscissa, including the pure spectral abscissa, is much more challenging. The authors' theoretical results on regularity of this function may be found in [BO01, BLO03a, Lew02].

Our MATLAB implementation of the GS algorithm is freely available. Furthermore, it is our intention to make publicly available a nonsmooth, nonconvex optimization test set that will include all the problems described here, so that others may use them in the future. For all its power, the GS algorithm is nothing more than a generalized steepest descent method, and will hopefully provide a benchmark against which other algorithms, perhaps more rapidly convergent and efficient, may be compared in the future.

REFERENCES

[AL04] K. ANSTREICHER AND J. LEE. *A masked spectral bound for maximum-entropy sampling*, in MODA 7—Advances in Model-Oriented Design and Analysis, A. di Bucchianico, H. Läuter, and H. P. Wynn, eds., Springer-Verlag, Berlin, 2004, pp. 1–10.

[BB90] S. BOYD AND V. BALAKRISHNAN, *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm*, Systems Control Lett., 15 (1990), pp. 1–7.

[BGL95] V. BLONDEL, M. GEVERS, AND A. LINDQUIST, *Survey on the state of systems and control*, Eur. J. Control, 1 (1995), pp. 5–23.

[BLO01] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Optimizing matrix stability*, Proc. Amer. Math. Soc., 129 (2001), pp. 1635–1642.

[BLO02a] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Approximating subdifferentials by random sampling of gradients*, Math. Oper. Res., 27 (2002), pp. 567–584.

[BLO02b] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Two numerical methods for optimizing matrix stability.* Linear Algebra Appl., 351–352 (2002), pp. 117–145.

[BLO03a] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Optimization and pseudospectra, with applications to robust stability*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 80–104.

[BLO03b] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Robust stability and a criss-cross algorithm for pseudospectra*, IMA J. Numer. Anal., 23 (2003), pp. 359–375.

[BO01] J. V. BURKE AND M. L. OVERTON, *Variational analysis of non-Lipschitz spectral functions*, Math. Program., 90 (2001), pp. 317–352.

[Bur85] J. V. BURKE, *Descent methods for composite nondifferentiable optimization problems*, Math. Program., 33 (1985), pp. 260–279.

[Bye88] R. BYERS, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 875–881.

[CG78] R. CHANEY AND A. GOLDSTEIN, *An extension of the method of subgradients*, in Nonsmooth Optimization, C. Lemaréchal and R. Mifflin, eds., Pergamon Press, Oxford, UK, New York, 1978, pp. 51–70.

[Cla73] F. H. CLARKE, *Necessary Conditions for Nonsmooth Problems in Optimal Control and the Calculus of Variations*, Ph.D. thesis, University of Washington, Seattle, WA, 1973.

[Cla83] F. H. CLARKE, *Optimization and Nonsmooth Analysis*, Classics Appl. Math. 5, SIAM, Philadelphia, 1990.

[CLSW98] F. H. CLARKE, YU. S. LEDYAEV, R. J. STERN, AND P. R WOLENSKI, *Nonsmooth Analysis and Control Theory*, Grad. Texts in Math. 178, Springer-Verlag, New York, 1998.

[Dav90] E. J. DAVISON, ED., *Benchmark Problems for Control System Design: Report of the IFAC Theory Committee*, Pergamon Press, Oxford, UK, 1990.

[DR95] V. F. DEMYANOV AND A. RUBINOV, *Constructive Nonsmooth Analysis*, Approx. Optim. 7, Peter Lang, Frankfurt am Main, Germany, 1995.

[FGG02] A. FUDULI, M. GAUDIOSO, AND G. GIALLOMBARDO, *A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization*, Optim. Methods Softw., 19 (2004), pp. 89–102.

[Fle87]   R. FLETCHER, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester, UK, New York, 1987.

[FM68]   A. V. FIACCO AND G. P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Classics Appl. Math. 4, SIAM, Philadelphia, 1990.

[Gol77]   A. A. GOLDSTEIN, *Optimization of Lipschitz continuous functions*, Math. Program., 13 (1977), pp. 14–22.

[Gro02]   A. GROTHEY, *A Second Order Trust Region Bundle Method for Nonconvex Nonsmooth Optimization*, Technical report MS02-001, University of Edinburgh, Edinburgh, UK, 2002.

[HUL93]   J. B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Convex Analysis and Minimization Algorithms* II: *Advanced Theory and Bundle Methods*, Springer-Verlag, Berlin, 1993.

[Kiw85]   K. C. KIWIEL, *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Math. 1133, Springer-Verlag, Berlin, New York, 1985.

[Lem75]   C. LEMARÉCHAL, *An extension of Davidon methods to non-differentiable problems*, in Nondifferentiable Optimization, M. L. Balinski and P. Wolfe, eds., Math. Programming Stud., 3 (1975), pp. 95–109.

[Lew99]   A. S. LEWIS, *Nonsmooth analysis of eigenvalues*, Math. Program., 84 (1999), pp. 1–24.

[Lew02]   A. S. LEWIS. *Robust regularization*, Math. Program., submitted; available at http://www.orie.cornell.edu/~aslewis/publications/Robust.ps.

[LSB91]   C. LEMARÉCHAL, J.-J. STRODIOT, AND A. BIHAIN, *On a bundle algorithm for nonsmooth optimization*, in Nonlinear Programming 4, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1991, pp. 245–282.

[LV98]   L. LUKŠAN AND J. VLČEK, *A bundle Newton method for nonsmooth unconstrained minimization*, Math. Program., 83 (1998), pp. 373–391.

[Mei67]   G. MEINARDUS, *Approximation of Functions: Theory and Numerical Methods*, Springer Tracts Nat. Philos. 13, Springer-Verlag, New York, 1967.

[MN92]   M. M. MÄKELÄ AND P. NEITTAANMÄKI, *Nonsmooth Optimization*, World Scientific, River Edge, NJ, 1992.

[Mos03]   MOSEK *Optimization Software*, http://www.mosek.com/, 2003.

[OKZ98]   J. OUTRATA, M. KOCVARA, AND J. ZOWE, *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*, Nonconvex Optim. Appl. 28, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[Osb85]   M. R. OSBORNE, *Finite Algorithms in Optimization and Data Analysis*, John Wiley, Chichester, UK, New York, 1985.

[Roc70]   R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[RW98]   R. T. ROCKAFELLAR AND R. J. B. WETS, *Variational Analysis*, Springer-Verlag, New York, 1998.

[SZ92]   H. SCHRAMM AND J. ZOWE, *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM J. Optim., 2 (1992), pp. 121–152.

[Tre97]   L. N. TREFETHEN, *Pseudospectra of linear operators*, SIAM Rev., 39 (1997), pp. 383–406.

[Wil97]   J. C. WILLEMS, *Generic eigenvalue assignability by real memoryless output feedback made simple*, in Communications, Computation, Control and Signal Processing, A. Paulraj, V. Roychowdhury, and C. D. Schaper, eds., Kluwer Academic Publishers, Norwell, MA, 1997, pp. 343–354.

[Wol75]   P. WOLFE, *A method of conjugate subgradients for minimizing nondifferentiable functions*, in Nondifferentiable Optimization, M. L. Balinski and P. Wolfe, eds., Math. Programming Stud., 3 (1975), pp. 145–173.

[Won85]   W. M. WONHAM. *Linear Multivariable Control: A Geometric Approach*, 3rd ed., Appl. Math. 10, Springer-Verlag, New York, 1985.

[Wri02]   T. G. WRIGHT, *EigTool: A graphical tool for nonsymmetric eigenproblems*, OxfordUniversity Computer Laboratory, Oxford, UK, http://web.comlab.ox.ac.uk/projects/pseudospectra/eigtool/ (2002).