A Semismooth Conjugate Gradients Method — Theoretical Analysis

Franz Bethke   Andreas Griewank   Andrea Walther

January 13, 2023

In large scale applications, deterministic and stochastic variants of Cauchy's steepest descent method are widely used for the minimization of objectives that are only piecewise smooth. In this paper we analyse a deterministic descent method based on the generalization of rescaled conjugate gradients proposed by Philip Wolfe in 1975 for objectives that are convex. Without this assumption the new method exploits semismoothness to obtain pairs of directionally active generalized gradients such that it can only converge to Clarke stationary points. Numerical results illustrate the theoretical findings.

Keywords: Generalized gradient, Semismoothness, Conjugate gradient method, Shortest residual update

1 Introduction and motivation

For the task of unconstrained minimization of a continuous nonsmooth function $f\colon \mathbb{R}^n \to \mathbb{R}$, there are only few algorithmic options, when one explicitly wants to exploit the nonsmoothness. The two main classes consist of methods based on subgradients [1] in the convex case and bundle methods [27] in the general Lipschitzian case, respectively. All these methods rely on the so-called oracle paradigm [13], namely the assumption that at any $x \in \mathbb{R}^n$ the user can provide the algorithm with at least one generalized gradient $g \in \partial_C f(x)$. This holds also true for the approach very close to the current proposal that was suggested and analyzed in the seminal paper [37] by Phil Wolfe for the convex case. While the oracle paradigm is usually described as a natural and reasonable requirement, we have argued in [16] that it is only realistic when the objective is given and analyzed as a piecewise differentiable function defined by an abs-smooth straight-line program and in that case this structure can be exploited much more effectively. Therefore, the new approach proposed in this paper exploits more structural information in that it is based on directionally active gradients. Furthermore, the subgradient and bundle methods depend on a significant number of hyper parameters, which need to be selected to reach at least stationary points at a reasonable rate. Probably for this reason, neither type of method has been included in software packages or is used widely in machine learning. For an introduction and an up to date survey of the field see [3] and [4].

Some authors [23] have advocated the use of quasi-Newton methods like BFGS with an adapted line-search for the nonsmooth case. There, the matrix approximating curvature necessarily tends to infinity in cross-kink directions and correspondingly its inverse becomes singular. Eventually, one encounters insurmountable numerical problems and the convergence theory for this approach is very limited, even in combination with gradient sampling [5]. Moreover, on large scale problems, also a limited memory version of this approach will have to deal with matrices of significant size. Again the size of the memory and other hyper parameters are crucial and not easy to choose.

Finally, there is a strand of articles based on the so-called proximal point method, which is in general more a theoretical construction than a practical algorithm. Only for special classes [24] the local proximal model problems can be solved efficiently. In general, they are computationally almost as hard to minimize as the original nonsmooth objective. A theoretical advantage is that the resulting iterates can be shown to cluster at *critical* or *first order minimal* rather than just Clarke stationary points.

The same is true for the methodology of successive abs-linearization (SALMIN) proposed and analyzed in [18] and [17]. Here, the local model problems are nonconvex piecewise linear problems, which can be solved exactly by an active signature method generalizing active set strategies for quadratic optimization [26]. We continue work on an efficient implementation of SALMIN, where *active kink* Jacobians are accumulated and factorized explicitly. In this approach no nonsmoothness, i.e., kink, is simply stepped over, which makes sense if there are not too many of them. Hence, the SALMIN approach is suitable for small to medium size problems.

To overcome this limitation, the approach proposed here avoids to stop at any kink. Furthermore, the main emphasis is on limiting the number of hyper parameters to an absolute minimum, so that the algorithm can be cast in autonomous software and numerical results can be recorded and replicated unambiguously by anyone using it. That should also simplify the task of software maintenance and the training and consultation of users. Our method has the classical structure of a successive descent iteration

$$x_{k+1} = x_k + \eta_k d_k \quad \text{with} \quad \eta_k = \operatorname*{argmin}_{\eta \in \mathbb{R}} f(x_k + \eta d_k),$$

where the directions d_k are updated based on convex combinations of generalized gradients. The step multipliers η_k are determined by a bracketing line-search that is much more accurate than usually recommended in nonsmooth optimization.

The paper is organized as follows. In the following Section 2 we review the basic concepts of nonsmooth analysis for directionally differentiable and semismooth functions, respectively. In Section 3 we describe our method from a mathematical point of view without regard for its numerical implementation. In Section 4 we derive some basic properties and prove that the method can only converge to Clarke stationary points. Section 5 reports preliminary numerical results before the article ends with a summary and outlook in Section 6.

To derive the presented results, we use the following notation. For $0 < \rho \in \mathbb{R}$, $n \in \mathbb{N}$ and $x \in \mathbb{R}^n$ the open ball around x of radius ρ is denoted by $B_\rho(x)$ and for any $M \subseteq \mathbb{R}^n$ we write $\overline{M}$ for the topological closure of M. For a function $f\colon \mathbb{R}^n \to \mathbb{R}$ the set of Fréchet-differentiable points is written as $\mathcal{D}_f \subseteq \mathbb{R}^n$ and the level set with respect to x as $\mathcal{L}_f(x) \coloneqq \{y \in \mathbb{R}^n \colon f(y) \le f(x)\}$.

2 Required constructs of nonsmooth analysis

Throughout it is assumed that the objective $f\colon \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz continuous, i.e., for every $x \in \mathbb{R}^n$ there is some $\rho \in (0, \infty)$ and $L \in [0, \infty)$ such that for all $y, z \in B_\rho(x)$

$$|f(y) - f(z)| \le L\|y - z\|.$$

The class of all locally Lipschitz continuous functions is denoted by $\mathcal{C}^{0,1}_{\text{loc}}(\mathbb{R}^n)$.

Generalized differentials and directional derivatives Under the above assumption of local Lipschitz continuity the set of Fréchet-differentiable points $\mathcal{D}_f$ of f is a dense subset of $\mathbb{R}^n$ by Rademacher [9, Theorem 2, p. 81]. Thus, for all $x \in \mathbb{R}^n$ the Bouligand differential

$$\partial_B f(x) \coloneqq \{g \in \mathbb{R}^n \colon \text{ there is } (x_k)_{k\in\mathbb{N}} \subset \mathcal{D}_f \text{ with } x = \lim_{k\to\infty} x_k \text{ and } g = \lim_{k\to\infty} \nabla f(x_k)\}$$

is nonempty and the Clarke differential can be characterized by

$$\partial_C f(x) \coloneqq \operatorname{conv}(\partial_B f(x)) \tag{1}$$

as shown in [7, Theorem 2.5.1]. The norm of the generalized gradients $g \in \partial_C f(x)$ is bounded by the Lipschitz constant L corresponding to x [7, Proposition 2.1.2].

Another consequence of the assumed local Lipschitz continuity of f on a finite dimensional Euclidean space is that virtually all definitions of directional derivatives are identical [33], except for the disoriented construction of Clarke [7] which yields for $x \mapsto |x|$ and $x \mapsto -|x|$ identical values at $x = 0$. In the same book it was shown that for $x, d \in \mathbb{R}^n$ the generalized directional derivative

$$f'(x; d) \coloneqq \lim_{\tau \searrow 0} \frac{f(x + \tau d) - f(x)}{\tau} \tag{2}$$

is continuous with respect to its second argument d. In the piecewise smooth case, $f'(x; d)$ is moreover piecewise linear with respect to d, see [32, Proposition 2.2.6.]. If $f'(x; d)$ exists for all $x, d \in \mathbb{R}^n$ the function f is said to be directionally differentiable. This is denoted by $f \in \mathcal{C}^1_{\text{dir}}(\mathbb{R}^n)$.

Directional differentiation is a linear operator with respect to function addition and scaling and it satisfies strict differentiation rules for products and nonlinear composition [34]. Directional derivatives can therefore be computed by the usual rules for composite functions from their constituents by way of automatic or algorithmic differentiation. As shown in [19] and cited in [20] that remains true even when the discontinuous sign function is allowed as elemental function. However, we will not pursue this generalization here.

Semismooth functions One classical analysis example function, see, e.g., [7, Example 2.2.3], is given by

$$f(x) = \begin{cases} x^2 \sin\bigl(x^{-1}\bigr) & \text{if } x \neq 0 \\ 0 & \text{else} \end{cases} . \tag{3}$$

This kind of function should not really appear in any practical computational model, if only because its numerical evaluation and especially its sign is dubious for reasonably small x. The function f given above is directionally differentiable but not in a strict sense, which leads to

$$\partial_B f(0) = \partial_C f(0) = [-1, 1] \qquad \text{but} \qquad f'(0; 1) = f'(0; -1) = 0 \; .$$

In other words, at the origin the Bouligand and Clarke differentials are not determined by the directional derivatives and thus not computable by chain rule based procedures. Hence, we make the following stronger assumption of semismoothness based on the definition given in [25].

Definition 1 (Semismoothness)**.** The locally Lipschitz continuous function $f\colon \mathbb{R}^n \to \mathbb{R}$ is called *semismooth* if for any $x, d \in \mathbb{R}^n$ and any sequences $(\tau_k)_{k\in\mathbb{N}} \subset \mathbb{R}$, $(\theta_k)_{k\in\mathbb{N}} \subset \mathbb{R}^n$ and $(g_k)_{k\in\mathbb{N}} \subset \mathbb{R}^n$ with

$$\tau_k \searrow 0, \qquad \theta_k/\tau_k \to 0 \qquad \text{and} \qquad g_k \in \partial_C f(x + \tau_k d + \theta_k) \; ,$$

the sequence $(\langle g_k, d\rangle)_{k\in\mathbb{N}}$ converges to the directional derivative $f'(x; d)$.

Loosely speaking the definition requires that the directional derivatives do not only exist, but that they vary continuously if the reference point x varies in a cusp like neighborhood. It is shown in [32, Proposition 3.1.2.] that the continuity of the directional derivative with respect to general variations of x in an open neighborhood already implies Fréchet differentiability and can thus not be required in the nonsmooth scanario.

The above example (3) is indeed not semismooth albeit differentiable. A standard example for a semismooth function is the Fischer-Burmeister complementarity function $f^{\mathrm{FB}}\colon \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f^{\mathrm{FB}}(x_1, x_2) \coloneqq \sqrt{x_1^2 + x_2^2} - x_1 - x_2 = \|(x_1, x_2)\| - x_1 - x_2 \; ,$$

as shown in [35]. In [25], Mifflin proved that semismoothness is maintained by linear combination, multiplication and composition. Hence, the semismooth functions form a linear space which we will denote by $\mathcal{C}^1_{\text{sem}}(\mathbb{R}^n)$. Therefore, the semismoothness of f^{FB} is equivalent to the semismoothness of the Euclidean norm $\operatorname{euc}(x_1, x_2) \coloneqq \sqrt{x_1^2 + x_2^2}$ and thus all functions that are compositions of euc and smooth elemental functions. The latter class will be denoted by $\mathcal{C}^1_{\text{euc}}(\mathbb{R}^n)$ and the subset of functions that uses only $\operatorname{abs}(x) \coloneqq \operatorname{euc}(x, 0) = |x|$ in their composition is denoted by $\mathcal{C}^1_{\text{abs}}(\mathbb{R}^n)$. A formal definition for the latter can be found in [36, Definition 2.2]. For $f \in \mathcal{C}^1_{\text{abs}}(\mathbb{R}^n)$, one may naturally calculate directional derivatives and generalized gradients by the forward and reverse application of the chain rule, respectively, see, e.g., [11]. This turns out to be still true when $f \in \mathcal{C}^1_{\text{euc}}(\mathbb{R}^n)$ as shown in [16]. In summary, we deal with the following hierarchy of assumptions:

$$\mathcal{C}^1_{\text{abs}}(\mathbb{R}^n) \subsetneq \mathcal{C}^1_{\text{euc}}(\mathbb{R}^n) \subsetneq \mathcal{C}^1_{\text{sem}}(\mathbb{R}^n) \subsetneq \mathcal{C}^1_{\text{dir}}(\mathbb{R}^n)$$

The property of semismoothness is used extensively in nonsmooth equation solving, where it guarantees superlinear convergence under the additional assumption of uniform invertibility of the generalized Jacobians at the solution point itself.

Directionally active gradients and first order minimality For the proposed method, it is important that one can pick gradients that are active in the following sense.

Lemma 1 (Directionally active gradients)**.** *For $f \in \mathcal{C}^1_{\text{sem}}(\mathbb{R}^n)$ and any $x, d \in \mathbb{R}^n$ the set of* directionally active gradients

$$\partial f(x; d) \coloneqq \{g \in \partial_B f(x) \colon \langle g, d\rangle = f'(x; d)\}$$

is nonempty.

Proof. For any two real sequences $(\tau_k)_{k\in\mathbb{N}}$, $(\varepsilon_k)_{k\in\mathbb{N}}$ with $\tau_k \searrow 0$ and $\varepsilon_k/\tau_k \searrow 0$ there is, by Rademacher's theorem, $x_k \in B_{\varepsilon_k}(x + \tau_k d) \cap \mathcal{D}_f$. Let $\theta_k \coloneqq x_k - (x + \tau_k d) \in B_{\varepsilon_k}(0)$ such that

$$\lim_{k\to\infty} \|\theta_k/\tau_k\| \le \lim_{k\to\infty} \varepsilon_k/\tau_k = 0 \ ,$$

and hence, $\theta_k/\tau_k \to 0$. Due to the local Lipschitz continuity of f the sequence of gradients $(g_k)_{k\in\mathbb{N}}, g_k \coloneqq \nabla f(x_k)$, must be bounded, and hence, it exhibits a converging subsequence $(g_{k_\ell})_{\ell\in\mathbb{N}}$ whose limit g consequently satisfies $g \in \partial_B f(x)$. Finally, the semismoothness of f implies that

$$\langle g, d\rangle = \lim_{\ell\to\infty} \langle g_{k_\ell}, d\rangle = f'(x; d). \qquad \square$$

A similar existence result for the smaller class of piecewise smooth functions is given in [22, Lemma 2.11]. Furthermore, a result for the infinite dimensional case can be found in [6, Lemma 4.5]. For the analysis of the algorithm proposed in this paper, we will use the following result.

Lemma 2. *For $f \in \mathcal{C}^1_{\text{sem}}(\mathbb{R}^n)$ and any $x, d \in \mathbb{R}^n$ $f'(x; d) = -\lim_{\tau\searrow 0} f'(x + \tau d; -d)$.*

Proof. For any monotonically declining zero sequence $(\tau_k)_{k\in\mathbb{N}}$ there exist directionally active gradients $g_k \in \partial f(x + \tau_k d; -d)$ such that

$$\lim_{k\to\infty} f'(x + \tau_k d; -d) = -\lim_{k\to\infty} \langle g_k, d\rangle = -f'(x; d) \ ,$$

where convergence follows from the semismoothness of f. $\square$

The univariate restriction $\phi(\eta) \coloneqq f(x + \eta d)$ of f in direction $d \in \mathbb{R}^n$ is semismooth due to the semismoothness of f. This allows for an explicit representation of its Clarke generalized derivative $\partial_C \phi$ in terms of its directional derivatives

$$\phi'_+(\eta) \coloneqq \phi'(\eta; 1) = f'(x + \eta d; d) \quad \text{and} \quad \phi'_-(\eta) \coloneqq -\phi'(\eta; -1) = -f'(x + \eta d; -d) \tag{4}$$

as proven next.

Lemma 3. *For $f \in \mathcal{C}^1_{\text{sem}}(\mathbb{R}^n)$ and $x, d \in \mathbb{R}^n$, define $\phi \in \mathcal{C}^1_{\text{sem}}(\mathbb{R})$, $\phi(\eta) := f(x + \eta d)$. Then, for all $\eta \in \mathbb{R}$*

$$\partial_B \phi(\eta) = \{\phi'_+(\eta), \phi'_-(\eta)\} \tag{5}$$

and thus

$$\partial_C \phi(\eta) = [\min(\phi'_-(\eta), \phi'_+(\eta)), \max(\phi'_-(\eta), \phi'_+(\eta))] \ , \tag{6}$$

which further reduces at almost all $\eta \in \mathbb{R}$ to the singleton

$$\partial_C \phi(\eta) = \{\phi'_-(\eta)\} = \{\nabla \phi(\eta)\} = \{\phi'_+(\eta)\} \ . \tag{7}$$

Proof. Let $g \in \partial_B \phi(\eta)$. Then there is a sequence $(\eta_k)_{k \in \mathbb{N}} \subset \mathcal{D}_\phi$ of Frèchet-differentiable points for ϕ that converges to η with $\nabla \phi(\eta_k) \to g$. The sequence $(\eta_k)_{k \in \mathbb{N}}$ contains a monotone subsequence $(\eta_{k_\ell})_{\ell \in \mathbb{N}}$ whose distance to the limit is denoted by $\tau_{k_\ell} := |\eta_{k_\ell} - \eta|$. If $(\eta_{k_\ell})_{\ell \in \mathbb{N}}$ is decreasing, the fact that $\eta_{k_\ell} \in \mathcal{D}_\phi$ and Lemma 2 show that

$$g = \lim_{\eta_{k_\ell} \searrow \eta} \nabla \phi(\eta_{k_\ell}) = \lim_{\eta_{k_\ell} \searrow \eta} -\phi'(\eta_{k_\ell}; -1) = \lim_{\tau_{k_\ell} \searrow 0} -\phi'(\eta + \tau_{k_\ell}; -1) = \phi'(\eta; 1) = \phi'_+(\eta) \ .$$

If otherwise $(\eta_{k_\ell})_{\ell \in \mathbb{N}}$ is increasing the same arguments show $g = \phi'_-(\eta)$. Thus,

$$\partial_B \phi(\eta) \subseteq \{\phi'_+(\eta), \phi'_-(\eta)\} \ .$$

Rademacher's theorem implies the existence of a sequence $(\eta_k)_{k \in \mathbb{N}} \subset \mathcal{D}_\phi$ with $\eta_k \searrow \eta$. Let $\tau_k := \eta_k - \eta > 0$ and $g_k := \nabla \phi(\eta + \tau_k)$. Then the semismoothness of ϕ implies that $g_k \to \phi'(\eta, 1) = \phi'_+(\eta)$ and hence $\phi'_+(\eta) \in \partial_B(\eta)$. If instead a sequence $(\eta_k)_{k \in \mathbb{N}} \subset \mathcal{D}_\phi$ with $\eta_k \nearrow \eta$ is chosen the same argument leads to $\phi'_-(\eta) \in \partial_B \phi(\eta)$ which completes the proof of Equation (5).

The other two statements (6) and (7) follow from the definition of the Clarke differential given in (1) and again Rademacher's theorem. □

Clearly, η can only be a local minimizer of ϕ if the first order condition

$$\phi'_-(\eta) \leq 0 \leq \phi'_+(\eta) \tag{8}$$

is satisfied. Therefore, a step size satisfying (8) will be called *first order minimal.* Note that in the smooth case, also a local maximizer satisfies this stationarity condition. From Equation (6), it follows that first order minimality is equivalent to

$$0 \in \partial_C \phi(\eta) = [\phi'_-(\eta), \phi'_+(\eta)] \ .$$

In our conceptual algorithm, we will assume that the line search algorithm satisfies this condition exactly and in an actual implementation still approximately as will be discussed in more detail in a companion paper.

3 The method in a nutshell

The proposed conjugate gradient method for semismooth objectives in Algorithm 1 consists of the following four main steps:

First, as typical for conjugate gradient methods, each iteration starts by computing a first order optimal point along a previously determined direction d by means of finding a suitable step size $\eta \in \mathbb{R}$. If d is a descent direction this can be achieved by the conceptual line search in Algorithm 3. If d happens to be no descent direction at x, but $-d$ is one, a line search will be performed in the opposite direction instead. This is encoded in the algorithm by possibly negative step sizes η. If neither d nor $-d$ give descent in the objective, Algorithm 1 performs a null-step, i.e., $\eta = 0$. All this is implemented in Algorithm 2 (fomin) and is assumed to produce exact first order minimal points along the line $\eta \mapsto x + \eta d$.

After the update of the iterate x, a generalized gradient $g \in \partial_C f(x)$, orthogonal to d, is determined by means of the function $\operatorname{ortho}(g_+, g_-; d)$ from a pair of directionally active gradients $g_+ \in \partial f(x; d)$ and $g_- \in \partial f(x; -d)$.

Lastly, the search direction $d \in \mathbb{R}^n$ is updated by choosing the norm minimal element in the convex combination of $-g$ and the previous direction d. This is encoded in the $\operatorname{short}(-g, d)$ function.

The algorithm uses the theoretical stopping criterion $d = 0$, which will only hold in the limit, except in very special circumstances or due to round off errors.

Algorithm 1 Semismooth Conjugate Gradient Method (sscg)

```
Require: $f \in \mathcal{C}^1_{\text{sem}}(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$, $\mathcal{L}_f(x_0)$ bounded, $d_0 \in \mathbb{R}^n \setminus \{0\}$
for $k = 1, 2, \dots$ do
    $\phi \leftarrow f(x_{k-1} + \bullet\, d_{k-1})$
    $\eta_k = \operatorname{fomin}(\phi)$
    $x_k = x_{k-1} + \eta_k d_{k-1}$                                    ▷ (11)
    $(g_+, g_-) \in \partial f(x_k; d_{k-1}) \times \partial f(x_k; -d_{k-1})$   ▷ (13)
    $g_k = \operatorname{ortho}(g_+, g_-; d_{k-1})$                         ▷ (15)
    $d_k = \operatorname{short}(-g_k, d_{k-1})$                             ▷ (16)
    if $d_k = 0$ then return $x_k$
```

No guaranteed descent In general, we cannot guarantee the condition that all directions d_k are down-hill in that $\phi'_+(0) = f'(x_k; d_k)$ as defined in Equations (2) and (4), respectively, is negative. Let alone do we require a so-called significant decrease as enforced by gradient related methods in the smooth case. When f happens to be convex or differentiable near x_k, Equation (16) implies that

$$f'(x_k; d_k) \leq \cos(\theta_k)^2 f'(x_k; -g_k) + \sin(\theta_k)^2 f'(x_k; d_{k-1}) \,.$$

In the differentiable case this relation holds as an equality with the first term on the right becoming simply $-\cos(\theta_k)^2 \|g_k\|^2$ and the second term vanishing due to the exact line search. In the convex case the inequality holds due to the sublinearity of the directional derivative $f'(x; d)$ with respect to d. However, even given convexity the negative of a generalized gradient does not need to be a descent direction, so that the first term may be nonnegative. Moreover, the second term will then definitely be positive at nonsmooth iterations since otherwise the exact line search along $-d_{k-1}$ would not have stopped at $x_k = x_{k-1} - \eta_k d_{k-1}$. Thus, we see that the chances for d_k to be a descent direction at x_k are somewhat doubtful, even in the convex case. Therefore, in the conceptual Algorithm 1, we allow for *null-steps* if $f'(x_k; -d_k) \geq 0$.

If x_k happens to be a Clarke stationary point it may happen that Alogorithm 1 performs infinitely many null-steps.

A semismooth bracketing line search procedure In the kth outer iteration of the sscg algorithm, the inner algorithm fomin finds a, possibly negative, step size η_k that satisfies the first order minimality conditions in (8) for $\phi(\eta) \coloneqq f(x_{k-1} + \eta d_{k-1})$ and $\phi(\eta_k) \leq \phi(0)$. This is achieved by performing the line search in Algorithm 3 either in direction d_{k-1}, $-d_{k-1}$ or a null-step, i.e., $\eta_k = 0$ as summarized in Corollary 1 at the end of this section.

The line search itself then considers a semismooth function $l \in \mathcal{C}^1_{\text{sem}}(\mathbb{R}_+)$ and assumes that $l'_+(0) < 0$. In case of descent in d_{k-1} for f, i.e, $f'(x_{k-1}; d_{k-1}) = \phi'_+(0) < 0$, the functions l and ϕ coincide. Otherwise, when $f'(x_{k-1}; -d_{k-1}) = -\phi_-(0) < 0$, the sign is flipped via $l(\tau) \coloneqq \phi(-\tau)$. Algorithm 3 searches for a $\tau_k > 0$ that satisfies

$$l(\tau_k) < l(0) \qquad \text{and} \qquad l'_-(\tau_k) \leq 0 \leq l'_+(\tau_k)\ , \tag{9}$$

by generating a nested sequence of intervals containing at least one suitable candidate.

Algorithm 2 First order minimal step size (fomin)

```
Require: $\phi \in \mathcal{C}^1_{\text{sem}}(\mathbb{R})$, $\mathcal{L}_\phi(0)$ bounded
if $\phi'_+(0) < 0$ then
    $l \leftarrow \phi$
    $\tau = \text{line\_search}(l)$
    $\eta = \tau$
else if $\phi'_-(0) > 0$ then
    $l \leftarrow \phi(-\bullet)$
    $\tau = \text{line\_search}(l)$
    $\eta = -\tau$
else
    $\eta = 0$
return $\eta$
```

Lemma 4. *Let $l \in \mathcal{C}^1_{\text{sem}}(\mathbb{R}_+)$, $0 < q < 1/2$ and $1 < Q$. Assume that $l'_+(0) < 0$ and that the level set $\mathcal{L}_l(0)$ is bounded. Then, Algorithm 3 stops at some $\tau_* > 0$ that satisfies* (9) *or it produces a converging sequence of nested intervals $(\check{\tau}_i, \hat{\tau}_i)_{i \in \mathbb{N}}$ with*

$$l'_+(\check{\tau}_i) < 0 \text{ and } l(\check{\tau}_i) \leq l(0) \qquad \text{and} \qquad 0 < l'_-(\hat{\tau}_i) \text{ or } l(\check{\tau}_i) \leq l(\hat{\tau}_i) \tag{10}$$

Algorithm 3 Line search

```
Require: $l \in \mathcal{C}^1_{\mathrm{sem}}(\mathbb{R}_+)$, $l'_+(0) < 0$, $0 < q < 1/2$, $1 < Q$, $\mathcal{L}_l(0)$ bounded
 1: $(\check{\tau}_0, \tau_0, \hat{\tau}_0) = (0, 1, \infty)$
 2: for $i = 1, 2, \ldots$ do
 3:     if $l(\tau_{i-1}) < l(\check{\tau}_{i-1})$ and $l'_-(\tau_{i-1}) \leq 0 \leq l'_+(\tau_{i-1})$ then        ▷ (9)
 4:         return $\tau_{i-1}$
 5:     if $l'_+(\tau_{i-1}) < 0$ and $l(\tau_{i-1}) < l(\check{\tau}_{i-1})$ then
 6:         $(\check{\tau}_i, \hat{\tau}_i) = (\tau_{i-1}, \hat{\tau}_{i-1})$
 7:     else        ▷ $0 < l'_-(\tau_{i-1})$ or $l(\check{\tau}_{i-1}) \leq l(\tau_{i-1})$
 8:         $(\check{\tau}_i, \hat{\tau}_i) = (\check{\tau}_{i-1}, \tau_{i-1})$
 9:     if $\hat{\tau}_i < \infty$ then
10:         $\tau_i \in [\check{\tau}_i + q(\hat{\tau}_i - \check{\tau}_i), \hat{\tau}_i - q(\hat{\tau}_i - \check{\tau}_i)]$
11:     else
12:         $\tau_i \in (Q\check{\tau}_i, \infty)$
```

that all contain some $\tau_* > 0$ satisfying (9)*.*

Proof. In the first part of the proof, we skip the interation index for better readability. Note that any new trial value τ is strictly between the lower bound $\check{\tau}$ and the upper bound $\hat{\tau}$ and thus the relation $\check{\tau} < \tau < \hat{\tau}$ is maintained during the iteration. Due to the boundedness of the level set $\mathcal{L}_l(0)$ and since $Q > 1$ the condition $\hat{\tau} < \infty$ will be satisfied after finitely many iterations and henceforth the length of the interval $[\check{\tau}, \hat{\tau}]$ is reduced in every iteration by the factor $1/2 < 1 - q < 1$. Moreover, the intervals are nested such that their infinite intersection is nonempty.

As long as the algorithm did not terminate the conditions in line 5 ensure on the one hand that the lower bounds $\check{\tau}$ retain the invariant $l'_+(\check{\tau}) < 0$ and on the other hand the strict monotonic decline of the values $l(\check{\tau})$. The latter implies the second asserted invariant $l(\check{\tau}) \leq l(0)$ where equality only holds for the initial $\check{\tau} = 0$. On the other hand, the upper bound $\hat{\tau}$ is only updated if the termination condition in line 3 did not hold but $0 \leq l'_+(\tau)$ or $l(\check{\tau}) \leq l(\tau)$, i.e., τ satisfies $0 < l'_-(\tau)$ or $l(\check{\tau}) \leq l(\tau)$. In summary, Equation (10) holds in every iteration.

Hence, the iteration generates a sequence of nested intervals $(\check{\tau}_i, \hat{\tau}_i)_{i \in \mathbb{N}}$ such that there exists some $\tau_{*,i} \in (\check{\tau}_i, \hat{\tau}_i)$ with $l(\tau_{*,i}) < l(\check{\tau}_i) \leq l(\check{\tau}_0)$ due to $l'_+(\check{\tau}_i) < 0$. Since $(\check{\tau}_i, \hat{\tau}_i)$ converge so does the sequence $(\tau_{*,i})_{i \in \mathbb{N}}$. Denoting the limit with τ_*, it follows that

$$l(\tau_*) = \lim_{i \to \infty} l(\tau_{*,i}) \leq l(\check{\tau}_1) \leq l(0) \ .$$

It remains to show that τ_* satisfies the second part of Equation (9), i.e., the first order optimality conditions. Since the sequence of lower bounds $(\check{\tau}_i)_{i \in \mathbb{N}}$ converges to τ_* it follows from Lemma 2 and the invariant for the lower bounds that

$$l'_-(\tau_*) = \lim_{i \to \infty} l'(\check{\tau}_i; 1) = \lim_{i \to \infty} l'_+(\check{\tau}_i) \leq 0 \ ,$$

where the sequence $\tau_* - \check{\tau}_i$ converges to 0 from above.

To apply the same argument for the other inequality in the first order optimality conditions, the sequence $(\hat{\tau}_i)_{i\in\mathbb{N}}$ cannot be used directly since $l'_-(\hat{\tau}_i) \geq 0$ does not hold necessarily for all $i \in \mathbb{N}$. For any i, where $l'_-(\hat{\tau}_i) < 0$, the already proven relations in Equation (10) yield $l(\hat{\tau}_i) \geq l(\check{\tau}_i) \geq l(\tau_*)$ as the values at the lower bounds converge to $l(\tau_*)$ from above. Thus, there must be an $\tilde{\tau}_i \in (\tau_*, \hat{\tau}_i)$ with $l'_-(\tilde{\tau}_i) \geq 0$. Choosing a monotone subsequence of

$$\hat{\hat{\tau}}_i := \begin{cases} \hat{\tau}_i & \text{if } l'_-(\hat{\tau}_i) \geq 0 \\ \tilde{\tau}_i & \text{else} \end{cases}$$

now allows to apply the same argument as before leading to $l'_+(\tau_*) \geq 0$ which concludes the proof. □

Corollary 1. *For $f \in \mathcal{C}^1_{\text{sem}}(\mathbb{R}^n)$ and $x_{k-1}, d_{k-1} \in \mathbb{R}^n$, define $\phi \in \mathcal{C}^1_{\text{sem}}(\mathbb{R})$, $\phi(\eta) := f(x_{k-1} + \eta d_{k-1})$. Assume that the level set $\mathcal{L}_\phi(0)$ is bounded. Then Algorithm 2 computes $\eta_k \in \mathbb{R}$ with*

$$\phi(\eta_k) \leq \phi(0) \qquad \textit{and} \qquad \phi'_-(\eta_k) \leq 0 \leq \phi'_+(\eta_k) .$$

Here equality of the function values only holds in the case of a null-step, i.e., $\eta_k = 0$.

Proof. If $\phi'_+(0) < 0$, Algorithm 2 uses $l = \phi$, $\eta_k = \tau_k$ and the assertion follows directly from Equation (9) ensured by Lemma 4. If $\phi'_-(0) > 0$, $l(\tau) = \phi(-\tau)$ which implies that

$$l'_+(\tau) = -\phi'_-(-\tau) \qquad \text{and} \qquad l'_-(\tau) = -\phi'_+(-\tau)$$

and $\eta_k = -\tau_k$ such that Equation (9) yields

$$-\phi'_+(\eta_k) = -\phi'_+(-\tau_k) = l'_-(\tau_k) \leq 0 \leq l'_+(\tau_k) = -\phi'_-(-\tau_k) = -\phi'_-(\eta_k)$$

and also $\phi(\eta_k) = l(\tau_k) < l(0) = \phi(0)$. Lastly, if $\phi'_-(0) \leq 0 \leq \phi'_+(0)$, Algorithm 2 chooses $\eta_k = 0$ and thus $\phi(\eta_k) = \phi(0)$. □

Introducing a momentum term Even in the twice continuously differentiable and strongly convex case, the exact discrete steepest descent method is known to approach the unique minimizer in a zig-zaging fashion that pushes the linear convergence rate arbitrarily close to 1, depending on the conditioning of the Hessian. To overcome this problem one introduces some inertia in the search direction by setting

$$x_k = x_{k-1} + \eta_k d_{k-1} \quad \text{with} \quad d_k = -\alpha_k g_k + \beta_k d_{k-1} \quad \text{and} \quad g_k \in \partial_C f(x_k) \cap \{d_{k-1}\}^\perp \tag{11}$$

for positive scalar parameters α_k and β_k.

In the classical conjugate gradient method for the smooth case, Fletcher and Reeves [12] recommended

$$\alpha_k^{\text{FR}} = 1, \quad \beta_k^{\text{FR}} = \|g_k\|^2/\|g_{k-1}\|^2 \quad \text{with} \quad g_k = \nabla f(x_k) \tag{12}$$

while computing η_k by a fairly exact line search so that $\langle g_k, d_{k-1}\rangle$ is nearly zero. If f is convex quadratic, β_k^{FR} ensures that the successive search directions d_{k-1} and d_k are conjugate with respect to the Hessian $A = \nabla^2 f \in \mathbb{R}^{n\times n}$ in that $\langle d_k, Ad_{k-1}\rangle = 0$. This algebraic conjugacy property then follows by induction for all pairs in the sequence of search directions until termination in at most n steps.

That concept of conjugacy still makes some sense for twice continuously differentiable objectives. For $f \in \mathcal{C}^1(\mathbb{R}^n)$, according to [37], two directions d_k and d_{k-1} are said to be conjugate with respect to f at x_{k-1} if

$$0 = \langle d_k, \nabla f(x_{k-1} + \eta_k d_{k-1}) - \nabla f(x_{k-1})\rangle \ .$$

There is no meaningful generalization to the semismooth case considered here, as far as we are aware. Nevertheless, even for the nonsmooth scenario that name makes some intuitive sense since the proposed method is just a rescaled version of the Fletcher Reeves *conjugate gradient method* in the smooth case as we will see below.

The orthogonal gradient selection In order to determine the next search direction we will use a pair of directionally active gradients

$$g_+ \in \partial f(x; d), \qquad g_- \in \partial f(x; -d) \tag{13}$$

at the updated iterate x and with respect to the direction d that was just used for the previous invocation of the algorithm fomin. With the resulting η of said invocation, Corollary 1 then guarantees that

$$\langle g_-, d\rangle = -f'(x; -d) = \phi'_-(\eta) \le 0 \le \phi'_+(\eta) = f'(x; d) = \langle g_+, d\rangle \ . \tag{14}$$

We define

$$g := \mathrm{ortho}(g_+, g_-; d) := \begin{cases} \dfrac{\langle g_+, d\rangle g_- - \langle g_-, d\rangle g_+}{\langle g_+, d\rangle - \langle g_-, d\rangle} & \text{if } \langle g_+, d\rangle \neq \langle g_-, d\rangle \\ \dfrac{g_+ + g_-}{2} & \text{else .} \end{cases} \tag{15}$$

In the first case of definition (15), i.e., $\langle g_-, d\rangle \neq \langle g_+, d\rangle$, it is easy to see that $\langle g, d\rangle = 0$ while (14) ensures that the coefficients $\langle g_+, d\rangle$ and $-\langle g_-, d\rangle$ are both nonnegative and thus $g \in \mathrm{conv}\{g_-, g_+\}$. In the second case of the definition the latter is clear while now (14) ensures $\langle g_-, d\rangle = 0 = \langle g_+, d\rangle$, and g_-, g_+ as well as their convex combination g are orthogonal to d. Due to the definition of the Clarke differential as the convex hull of the Bouligand differential, we have thus obtained a generalized gradient $g \in \partial_C f(x) \cap \{d\}^\perp$ that can replace the proper gradient in the direction update formulate of the classical smooth conjugate gradient method. Notice that Wolfe in [37] proposed to only use the generalized gradient g_+ on the far side of a kink.

When ϕ is differentiable at η the minimality condition (8) reduces to $\phi'_-(\eta) = \phi'_+(\eta) = 0$ in which case we can set g to any convex combination of g_+ and g_-. Moreover, it is then usually the case that f itself is differentiable at $x + \eta d$ so that the pair (g_+, g_-) reduces to the singleton $g_+ = g_- = g$.

The shortest convex direction update There are very many different approaches for defining the momentum term, see, e.g., [28] for an early reference, some of which are derived by discretizations of second order ordinary differential equations or inclusions. We choose the coefficients in Equation (11) such that the resulting new search direction d_k has minimal Euclidean norm subject to $\alpha_k \geq 0 \leq \beta_k$ and $\alpha_k + \beta_k = 1$. This definition has been called in [30] and [8] the *smallest residual* method, but that label is not all that descriptive and furthermore invites confusion with the much better known *minimal residual method* in numerical linear algebra [10].

Under the orthogonality assumption $\langle g_k, d_{k-1}\rangle = 0$ enforced by the line search, the new search direction d_k is defined by the simple relation

$$d_k \coloneqq \text{short}(-g_k, d_{k-1}) \coloneqq \frac{-\|d_{k-1}\|^2 g_k + \|g_k\|^2 d_{k-1}}{\|g_k\|^2 + \|d_{k-1}\|^2} = -\alpha_k g_k + \beta_k d_{k-1} , \tag{16}$$

where the coefficients α_k and β_k are given by

$$\alpha_k \coloneqq \frac{\|d_{k-1}\|^2}{\|g_k\|^2 + \|d_{k-1}\|^2} = \frac{\|d_{k-1}\|^2}{\|g_k + d_{k-1}\|^2} = \left(\frac{\langle d_{k-1}, g_k + d_{k-1}\rangle}{\|d_{k-1}\|\|g_k + d_{k-1}\|}\right)^2 = \cos(\theta_k)^2 ,$$
$$\beta_k \coloneqq \frac{\|g_k\|^2}{\|g_k\|^2 + \|d_{k-1}\|^2} = 1 - \alpha_k = \sin(\theta_k)^2 ,$$

and $\theta_k \in [0, \pi/2)$ is the angle between the vectors d_{k-1} and $(g_k + d_{k-1})$ as well as between d_k and $-g_k$ as depicted in Figure 1. In other words d_k will be the shortest convex combination of $-g_k$ and d_{k-1}. This is similar to the minimal set G_k considered by Wolfe for his method in [37]. Furthermore, it follows for $g_k = 0$ that $d_k = 0$ and hence Algorithm 1 stops.

The length of d_k is monotonially reduced according to

$$\|d_k\| = \cos(\theta_k)\|g_k\| = \sin(\theta_k)\|d_{k-1}\| \leq \|d_{k-1}\|. \tag{17}$$

As already observed by Wolfe in [37], it then follows by induction from $0 \neq d_0 = -g_0 \in \partial_C f(x_0)$ that

$$d_k \in -\operatorname{conv}\{g_0, g_1, \ldots, g_k\} \quad \text{for} \quad k = 0, 1 \ldots \tag{18}$$

so that we have in fact an extremely limited memory version of a bundle method.

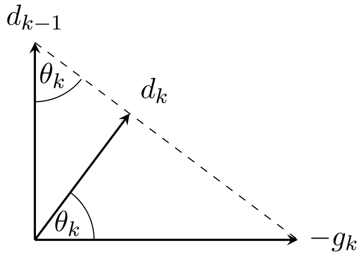


Figure 1: The shortest convex direction update.

4 Basic convergence properties

Relation to the Fletcher-Reeves conjugate gradients method When applied to positive definite quadratic objectives the proposed method represents of course a Krylov subspace iteration, and we found experimentally that it produces exactly the same iterates as the classical conjugate gradient method. Actually, Wolfe already made this observation almost 50 years ago [37] and he noted that the proposed method is just a rescaled version of the classical conjugate gradient method that was already discussed by Hestenes and Stiefel in [21]. More specifically, we have the following result for the general scenario $f \in \mathcal{C}^1_{\text{dir}}(\mathbb{R}^n)$.

Lemma 5 (Rescaled Fletcher Reeves)**.** *Let $f \in \mathcal{C}^1_{\text{sem}}(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$, and the level set $\mathcal{L}_f(x_0)$ be bounded. For $-g_0 := d_0^{\text{FR}} := d_0 \in \mathbb{R}^n \setminus \{0\}$, assume that the iteration of Algorithm 1 did not stop up to some $K \in \mathbb{N}$, i.e., $d_k \neq 0$ for all $k \leq K$. Then, the vectors $d_k^{\text{FR}} := \|g_k\|^2 \|d_k\|^{-2} d_k$ satisfy the recurrence*

$$d_k^{\text{FR}} = -g_k + \beta_k^{\text{FR}} d_{k-1}^{\text{FR}} = -\|g_k\|^2 \sum_{j=0}^{k} \frac{g_j}{\|g_j\|^2}, \tag{19}$$

for $1 \leq k \leq K$, which means that they are identical to the directions generated by the Fletcher-Reeves method in Equation (12).

Proof. Note that $d_k \neq 0$ implies $g_k \neq 0$ and thus from the definition of d_k^{FR} and the update formula of the d_k in Equation (16) it follows

$$d_k^{\text{FR}} = \frac{\|g_k\|^2 d_k}{\|d_k\|^2} = \frac{\|g_k\|^2 \|d_{k-1}\|^2 \left(-g_k + \|d_{k-1}\|^{-2} \|g_k\|^2 d_{k-1}\right)}{\|d_k\|^2 (\|g_k\|^2 + \|d_{k-1}\|^2)} \ .$$

Using the definition of $\sin(\theta_k)^2$ and the squared form of (17) we see that

$$\frac{\|g_k\|^2 \|d_{k-1}\|^2}{\|g_k\|^2 + \|d_{k-1}\|^2} = \|d_{k-1}\|^2 \sin(\theta_k)^2 = \|d_k\|^2$$

such that

$$d_k^{\text{FR}} = -g_k + \frac{\|g_k\|^2}{\|d_{k-1}\|^2} d_{k-1} = -g_k + \frac{\|g_k\|^2}{\|g_{k-1}\|^2} d_{k-1}^{\text{FR}} = -g_k + \beta_k^{\text{FR}} d_{k-1}^{\text{FR}}$$

as asserted. The second part follows from induction. □

This is exactly the nonlinear conjugate gradient version of Fletcher and Reeves [12] given in Equation (12) with the gradients g_k being identical for both formulations in the smooth case. Notice that we have not exploited the particular properties of the conjugate gradient method in the quadratic or even just smooth case. The ratio $\|g_k\|/\|d_k\|$ is likely to be bounded in the smooth case but will blow up in the nonsmooth case, where the generalized gradients $\|g_k\|$ are typically bounded away from zero but $d_k = O(1/\sqrt{k})$ tends to zero according to the bound (23) derived in the following section. Thus, the length of the Fletcher-Reeves direction $\|d_k^{FR}\| = \|g_k\|^2/\|d_k\|$ will grow towards infinity, which does not seem a good idea for the actual line search procedure. The first observation in the following section also suggests that the re-scaling is more natural in the nonsmooth context. For the Fletcher-Reeves method, one has in the smooth case under suitable assumptions the global convergence result that

$$\liminf_{k \to \infty} \|\nabla f(x_k)\| = 0 \ ,$$

see, e.g., [29] for a result with an exact line search or [2], [26, Theorem 5.7] for the case of a strong Wolfe line search. However, there is no convergence result regarding the function values or the iterates themself. Hence, we can not expect to get stronger results in the nonsmooth case.

As in the proof of Lemma 5 we obtain from the squared version of Equation (17)

$$\frac{1}{\|d_k\|^2} = \frac{\|g_k\|^2 + \|d_{k-1}\|^2}{\|g_k\|^2\|d_{k-1}\|^2} = \frac{1}{\|d_{k-1}\|^2} + \frac{1}{\|g_k\|^2} = \sum_{j=0}^{k} \frac{1}{\|g_j\|^2} \,, \tag{20}$$

which has been used widely in the literature in one form or another, see e.g., [29, 2]. From Equation (19) multiplied by $\|g_k\|^{-2}\|d_k\|^2$, one obtains

$$d_k = -\|d_k\|^2 \sum_{j=0}^{k} \frac{g_j}{\|g_j\|^2} \,. \tag{21}$$

This together with the reciprocal of (20) shows that indeed d_k is in the negative convex hull of all previous generalized gradients as already noted in Equation (18). In the nonsmooth case considered here, we likely have

$$0 < \gamma_0 := \inf_k \|g_k\| \le \sup_k \|g_k\| =: \gamma_1 < \infty \,, \tag{22}$$

where the uniform upper bound is ensured by our assumption of local Lipschitz continuity for f if the iteration is started at x_0 within a compact level set $\mathcal{L}_f(x_0)$. That means the averaging of the generalized gradients in Equation (21) is more or less uniform. We obtain from Equation (20) that

$$\frac{\gamma_0^2}{k+1} \le \|d_k\|^2 \le \frac{\gamma_1^2}{k+1} \,, \tag{23}$$

and further with Equations (16), (22) and the *condition number* $\kappa := \gamma_1/\gamma_0 \ge 1$ that

$$\frac{1}{1+k\kappa^2} \le \cos(\theta_k)^2 \le \frac{1}{1+k/\kappa^2} \,.$$

Thus, we see that the so-called Zoutendijk conditon [38] is exactly satisfied. More specifically, the search directions $-d_k$ and the negative gradients $-g_k$ do not become orthogonal too fast in that the squared cosines of the angles between them are just not summable. When f is Lipschitz continuously differentiable, one can guarantee by one of several so-called efficient line searches that when started from any $x_0 \in \mathcal{L}_f(x_0)$ the gradients g_k cannot be bounded away from zero so that the sequence $(x_k)_{k\in\mathbb{N}}$ must have at least one stationary cluster point. In the nonsmooth scenario considered in this paper we obtain the following weaker result.

Theorem 1 (Selective convergence)**.** *Let $f \in \mathcal{C}^1_{\text{sem}}(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$, $d_0 \in \mathbb{R}^n \setminus \{0\}$ and assume that the level set $\mathcal{L}_f(x_0)$ is bounded. If the iteration of Algorithm 1 stops at some index $K \in \mathbb{N}$, then x_K is a Clarke stationary point of f, i.e., $0 \in \partial_C f(x_K)$. Otherwise, if the iteration sequence $(x_k)_{k\in\mathbb{N}}$ has a limit x_* it must be a Clarke stationary point. Furthermore, the number of consecutive null-steps at any non-limit point is finite.*

Proof. If the iteration of Algorithm 1 stopped at x_K then $d_K = 0$ and $d_{K-1} \neq 0$. This however can only be due to $0 = g_K \in \partial_C f(x_K)$ in the short update rule in Equation (16).

For the second case, let some $\varepsilon > 0$ be given. Because of the upper semicontinuity of the Clarke differential [7, Proposition 2.1.5] and the assumed convergence of x_k there is a $\delta > 0$ and a $\tilde{k} \in \mathbb{N}$ such that $x_k \in B_\delta(x_*)$ for all $k > \tilde{k}$ and

$$\partial_C f(x_k) \subset \partial_C f(x_*) + \overline{B_\varepsilon(0)} \, .$$

Note that the latter set is a sum of convex and compact sets, hence itself is convex and compact. For any choices of $g_k \in \partial_C f(x_k)$ define the convex combination

$$\tilde{g}_k := \frac{\|d_{\tilde{k}}\|^2 \|d_k\|^2}{\|d_{\tilde{k}}\|^2 - \|d_k\|^2} \sum_{j=\tilde{k}+1}^{k} \frac{g_j}{\|g_j\|^2} \, ,$$

where the factor in front of the sum ensures that the coefficients of $\tilde{g}_k$ with respect to the g_j really sum to 1 because of (20). Using (21) twice, first for $\tilde{k}$ and then for k it follows

$$\|d_k\|^2 \sum_{j=\tilde{k}+1}^{k} \frac{g_j}{\|g_j\|^2} = \|d_k\|^2 \left(\sum_{j=0}^{k} \frac{g_j}{\|g_j\|^2} + \frac{d_{\tilde{k}}}{\|d_{\tilde{k}}\|^2} \right) = -d_k + \frac{\|d_k\|^2}{\|d_{\tilde{k}}\|^2} d_{\tilde{k}},$$

and hence,

$$\tilde{g}_k = \left(-d_k + \frac{\|d_k\|^2}{\|d_{\tilde{k}}\|^2} d_{\tilde{k}} \right) \frac{\|d_{\tilde{k}}\|^2}{\|d_{\tilde{k}}\|^2 - \|d_k\|^2} \, .$$

Since we know from (23) that $d_k \to 0$ it follows also that $\tilde{g}_k \to 0 \in \partial_C f(x_*) + \overline{B_\varepsilon(0)}$ and thus finally that $0 \in \partial_C f(x_*)$ as we may choose ε arbitrarily small.

As we have noted before the iteration can only become finite if for some $d_K = 0 = g_K$ for some K in which case $x_K = x_*$ is clearly Clarke stationary. If infinitely many null-steps happen at some x_k in a row this point is the limit and thus must be Clarke stationary as was proven before. □

While the iteration cannot get stuck at a single nonstationary point, however it is theoretically possible that the search path descents gradually along a downward spiral to some sort of limiting orbit. We have yet to observe this behavior in practice.

One might assume that in actual computations it would be very unlikely that the iterates are attracted to a set that does not even contain a stationary point. In our limited numerical experience, for example on piecewise linear convex or nonconvex problems, that never happened, possibly also due to perturbations by round-off errors. Therefore, we have so far refrained from adding additional safeguards, which might improve the global convergence theory, but would make the method more complicated and thus less elegant.

From a theoretical point of view, we obtain the following fairly strong but not entirely satisfactory convergence result for the general nonconvex case:

Theorem 2. *Let $f \in \mathcal{C}^1_{\mathrm{sem}}(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$, $d_0 \in \mathbb{R}^n \setminus \{0\}$ and assume that the level set $\mathcal{L}_f(x_0)$ is bounded. Then, if the iteration in Algorithm 1 does not stop, there is a nonempty set of cluster points X_* and a value $f_* \in \mathbb{R}$ such that*

$$0 \in \operatorname{conv}\left\{ \bigcup_{x_* \in X_*} \partial_C f(x_*) \right\} \qquad \textit{and} \qquad X_* \subseteq f^{-1}(f_*) \; . \tag{24}$$

When $X_* = \{x_*\}$ is a singleton the point x_* is Clarke stationary and if furthermore f is convex near x_* we must have a local minimizer.

Proof. Due to the nonascent condition, $f(x_k) \leq f(x_{k-1})$ for all $k \in \mathbb{N}$, ensured by the linesearch, the iterates must belong to the compact level set $\mathcal{L}_f(x_0)$. Thus, the set of cluster points X_* is nonempty and closed.

If for some $k \in \mathbb{N}$ one has that the generalized gradient $g_k = 0$ it follows that $d_k = 0$ by the short update rule in Equation (16) in contradiction to the assumption that the iteration did not stop. Thus, $g_k \neq 0$ for all $k \in \mathbb{N}$.

If $(g_k)_{k\in\mathbb{N}}$ exhibit a subsequence that converges to 0 the corresponding subsequence of iterates x_k must have a cluster point x_*. That x_* is a Clarke stationary point due to the outer semicontinuity of $\partial_C f$. Then the assertion holds as zero is contained in the set $\partial_C f(x_*)$ and thus its union with other generalized differentials.

Thus, the only case left is $0 < \gamma_0 < \gamma_1 < \infty$ such that Equation (23) holds. Now let, for $k \in \mathbb{N}$,

$$\delta_k := \sup_{j \geq k} \min_{x_* \in X_*} \|x_j - x_*\| \; ,$$

where the existence of the minimum is justified by the closedness of X_*. Assume that $(\delta_k)_{k\in\mathbb{N}}$ would not converge to 0. That is, there is a constant $c > 0$ such that for all $k \in \mathbb{N}$ there is $j_k \in \mathbb{N}$ with $j_k \geq k$ and

$$\min_{x_* \in X_*} \|x_{j_k} - x_*\| > c \; .$$

However, the bounded sequence $(x_{j_k})_{k\in\mathbb{N}}$ itself must have a cluster point $\tilde{x}_* \in X_*$ contradicting the existence of such a c and thereby proving that $\delta_k \to 0$ as $k \to \infty$. That means, for all $\delta \in (0, \infty)$ there is an index $k \in \mathbb{N}$ such that for all $j \in \mathbb{N}$ with $j \geq k$

$$\|x_j - x_{*_j}\| < \delta \; ,$$

where $x_{*_j} \in \operatorname{argmin}_{x^* \in X_*} \|x_j - x_*\|$. The upper semicontinuity of $\partial_C f$ then translates the difference in the input space to a difference in the gradients, i.e., for all $\varepsilon \in (0, \infty)$ there is an index $k \in \mathbb{N}$ such that for all $j \geq k$

$$\sup_{g_j \in \partial_C f(x_j)} \inf_{g_{*_j} \in \partial_C f(x_{*_j})} \|g_j - g_{*_j}\| < \varepsilon \; .$$

Taking a specific choice for the g_j in the supremum, namely those generated by the algorithm in iteration j and enlarging the set of the infimum to $G_* := \bigcup_{x_* \in X_*} \partial_C f(x_*) \supset \partial_C f(x_{*_j})$, for all j, then leads to

$$\inf_{g_* \in G_*} \|g_j - g_*\| < \varepsilon \; .$$

Taking the supremum over $j \geq k$ it follows that for each $\varepsilon > 0$ there is a $k_\varepsilon \in \mathbb{N}$ with

$$\sup_{k \geq k_\varepsilon} \inf_{g_* G_*} \|g_k - g_*\| \leq \varepsilon \ ,$$

i.e., $\operatorname{dist}(g_k, G_*) \coloneqq \inf_{g_* \in G_*} \|g_k - g_*\| \leq \varepsilon$ for all $k \geq k_\varepsilon$. It now follows from Equation (21) that

$$d_k = \frac{\|d_k\|^2}{\|d_{k_\varepsilon - 1}\|^2} d_{k_\varepsilon - 1} - \|d_k\|^2 \sum_{j=k_\varepsilon}^{k} \frac{g_j}{\|g_j\|^2}$$

and from Equation (17), by induction, that

$$\|d_k\| = \|d_0\| \prod_{j=1}^{k} \sin(\theta_j) \quad \text{and} \quad \beta_{k_\varepsilon, k} \coloneqq \prod_{j=k_\varepsilon}^{k} \beta_j = \prod_{j=k_\varepsilon}^{k} \sin(\theta_j)^2 = \frac{\|d_k\|^2}{\|d_{k_\varepsilon - 1}\|^2} \ .$$

Combining the above then yields

$$d_k \in (1 - \beta_{k_\varepsilon, k}) \operatorname{conv}\{-g_{k_\varepsilon}, \ldots, -g_k\} + \beta_{k_\varepsilon, k} d_{k_\varepsilon - 1} \ ,$$

where Equation (23) implies that $\beta_{k_\varepsilon, k}$ converge to zero as k tends to infinity for any fixed k_ε. Hence, we conclude that

$$\begin{aligned} 0 &= \lim_k \operatorname{dist}(d_k, \operatorname{conv}\{-g_{k_\varepsilon}, \ldots, -g_k\}) \\ &\geq \lim_k \operatorname{dist}(d_k, -\operatorname{conv} G_*) - \varepsilon \\ &= \operatorname{dist}(0, -\operatorname{conv} G_*) - \varepsilon \ , \end{aligned}$$

which proves the assertion (24) since ε can be chosen arbitrarily small. The final statement follows immediately. □

Hence, we could at least classify the cluster points of the generated iterates to some extend. If X_* has a reasonably small diameter one might interpret its elements as ε-stationary points in the sense of [14], but there is no guarantee for that. Furthermore, one might assume that in practice it would be very unlikely that the iterates are attracted to a level set $f^{-1}(f_*)$ that does not even contain a stationary point.

The theoretical properties derived in this section can be illustrated on the simple Euclidean norm example $f\colon \mathbb{R}^n \to \mathbb{R}$ with $n > 1$ defined by

$$f(x) = \|x\| \quad \text{with} \quad \nabla f(x) = g(x) = x/\|x\| \quad \text{if} \quad x \neq 0. \tag{25}$$

It is directionally but not piecewise differentiable or as we introduced in the first section in $\mathcal{C}^1_{\text{euc}}(\mathbb{R}^n)$ rather than abs-smooth. It can also be viewed as the limit of piecewise linear rotationally symmetric inverted pyramids with a number of faces tending to infinity. The corresponding optimization trajectories will then also converge to those of the limiting function $\|x\|$, which is everywhere differentiable except at the minimizer x_*. Hence, our method becomes standard conjugate gradient with an exact line search, except that we assume d_0 to be chosen arbitrarily. That can happen when f is defined differently outside a certain ball and the iteration reaches its inside at x_1 along an arbitrary direction d_0 from some fictitious point x_0, where we renumber the direction to exploit the identities derived in the Sections 3 and 4. Then, on the one hand, we obtain from $\|g_k\| = 1$ for $k > 0$ by induction from the recurrences (20) and (17)

$$\|d_k\|^2 = \frac{\|d_0\|^2}{1 + k\|d_0\|^2} \qquad \text{and} \qquad \sin(\theta_k)^2 = \frac{1 + (k-1)\|d_0\|^2}{1 + k\|d_0\|^2} \ ,$$

while, on the other hand, with $\|d_{k-1}\| = \cos(\theta_{k-1})$ it follows that

$$\langle x_k, d_{k-1}\rangle = \|x_k\|\langle g_k, d_{k-1}\rangle = 0 \quad \text{and} \quad \langle x_{k-1}, d_{k-1}\rangle = -\|x_{k-1}\|\cos(\theta_{k-1})^2,$$

and thus,

$$\|x_{k-1}\| = \frac{-\langle x_{k-1}, d_{k-1}\rangle}{\cos(\theta_{k-1})^2} = \frac{-\langle x_{k-1} + \eta_k d_{k-1}, d_{k-1}\rangle + \eta_k\|{d_{k-1}}^2\|}{\cos(\theta_{k-1})^2} = \eta_k.$$

Moreover, it then follows geometrically that

$$f(x_k)^2 = \|x_k\|^2 = \|x_{k-1}\|^2\sin(\theta_{k-1})^2 = \|x_1\|^2/k.$$

Hence, $\|d_k\|$, $\cos(\theta_k)$, $f(x_k) - f(x_*) = f(x_k)$, and $\|x_k - x_*\|$ all decline indeed like $1/\sqrt{k}$ on this example such that we have indeed convergence also of the function values and the iterates. It is interesting that the speed of convergence of the function values and iterates actually increases with the length $\|d_0\|$ of the original search direction. If it were infinity we would get $d_1 = g_1$ and one step of steepest descent would get us to the exact solution $x_2 = x_* = 0$. The smaller d_0 is the more reluctant the search trajectory adjusts to the local gradient g_k and the slower is the convergence.

5 Preliminary numerical results

To turn the conceptual ideas presented in the previous sections into an implementable algorithm we made to following changes. For the lines search termination criterion we used $\hat{\tau} - \check{\tau} < 10^{-13}$ and determined the directionally active gradients at the corresponding bounds $\check{\eta}, \hat{\eta}$ of the last iteration instead of at the theoretical minimizer η_k, i.e.,

$$g_{+,k} \in \partial f(x_{k-1} + \hat{\eta}_k d_{k-1}; d) \qquad \text{and} \qquad g_{-,k} \in \partial f(x_{k-1} + \check{\eta}_k d_{k-1}; -d) \ .$$

We used $Q = 2$ and $q = 8/9$ in Algorithm 3 and choose the new trial value τ_i by bisection of the interval $[\check{\tau}_i, \hat{\tau}_i]$. For the stopping of the sscg algorithm itself we simply fix the number of iterations to $K = 200$.

A preliminary implementation of the Algorithm 1 using MATLAB is tested for two examples showcasing different features. First we consider the large scale nonconvex and piecewise smooth test problem *chained crescent II* to demonstrate the performance for this adversarial class of problems.

As a second example we consider an optimization problem motivated by an application in image denoising and compare to the state of the art split Bregman algorithm, see [15]. In contrast to the first example this is a convex problem.

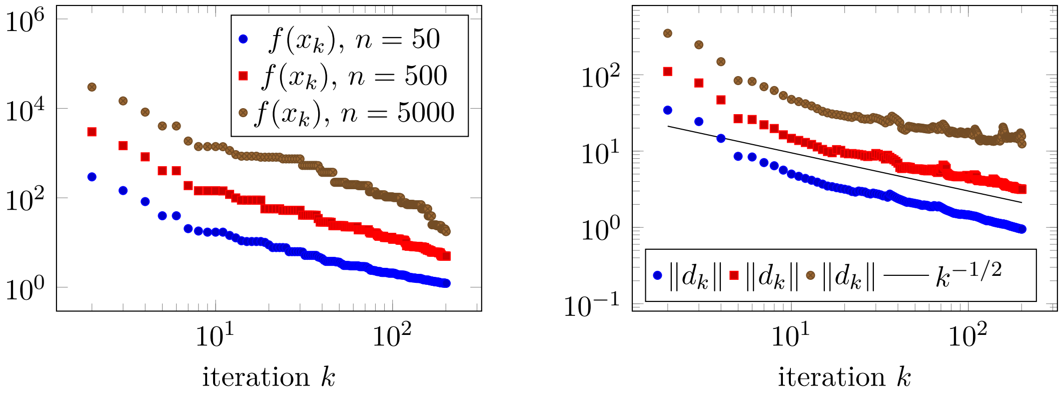
Chained crescent II This standard test problem from nonsmooth optimization, see, e.g., [3], has the form

$$f(x) \coloneqq \sum_{i=1}^{n-1} \max\left\{x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1, -x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right\},$$

$$x_{0,i} \coloneqq \begin{cases} -1.5 & \text{if } i \bmod 2 = 1 \\ 2 & \text{else .} \end{cases}$$

and is therefor nonconvex. The optimal function value $f(x_*) = 0$ is obtained in $x_* = (0, \dots, 0)$. In Figure 2a we report the observed function values over the iterations for three different choices of n, namely $n = 50$, $n = 500$ and $n = 5000$. We observe that for higher dimension we get longer spans of iterations in which the algorithm has to do nullsteps in order to find a new search direction, however we have no reason to believe that the overall reduction in the function values would break for longer computations.

Figure 2b shows the norm of the search direction $\|d_k\|$ that in all three cases converge roughly like the $k^{-1/2}$ as predicted by (23). We see however certain iterations in which $\|d_k\|$ is not reduced. This may be due to the not quite exact line search resulting in a small violation of the orthogonality of g_k and d_{k-1}. The problem also seems to arise more often if n increases.



(a) Function values at the iterates.

(b) Norm of search directions.

Figure 2: Convergence history plots for the chained crescent II example.

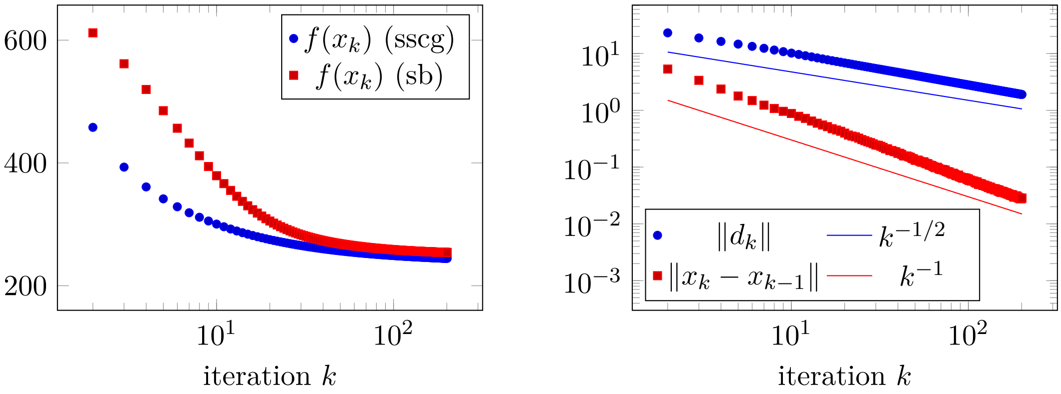
Image denoising As a second example we consider a classical application in image denoising using the popular ROF model, see [31], for the well known cameraman test image. For this the pixel values of an image are represented in vectors $x \in \mathbb{R}^n$, $n = 256^2$, and $D\colon \mathbb{R}^n \to \mathbb{R}^j$ is an operator that yields the $j = 2 \cdot 256 \cdot 255$ differences in neighboring pixels. The objective reads

$$f(x) = \tfrac{1}{2}\|x - x_d\|_2^2 + \rho\|Dx\|_1$$

for a positive parameter $\rho \in \mathbb{R}$ and a given noisy image x_d. Minimization of this objective aims to find an image x that is close to the given one while reducing the ℓ_1-norm of the jumps. For the target x_d the original image x_* is perturbed by adding normally distributed noise uniformly to all pixels of the original image, i.e., $x_{d,j} = x_{*,j} + \max(x_*)\mathcal{N}(0,1)/20$ for $j = 1, \ldots, n$.

In contrast to the nonconvex example we observe here no nullsteps and the norm of the search directions $\|d_k\|$ reduces monotonically and with the expected rate as depicted in Figure 3b. Moreover, the actual step length $\|x_k - x_{k-1}\|$ reduce linearly with the number of iterations k.

Most importantly, the sscg algorithm outperforms the split Bregman algorithm when compared over the number of iterations as can be seen in Figure 3a.



(a) Function values at the iterates.

(b) Norm of search directions.

Figure 3: Convergence history plots for the image denoising example.

6 Summary and outlook

We propose and implement a conjugate gradient based descent method that is well defined for semismooth problems and does not make any convexity assumption. The convergence analysis in Section 4 is rather simple and elegant but quite powerful, though certainly far from complete. We show in Theorem 1 that the method can only converge to Clarke stationary points within a compact level set.

The method relies on a fairly accurate bracketing line-search, which typically requires a handful of function and directional derivative evaluations. Per outer iteration one needs two directionally active gradient, which can be obtained in the reverse mode of algorithmic differentiation, a.k.a. back propagation, see, e.g., [11].

The method properties are a significant advance on the state of the art, especially since it automatically selects the step sizes and requires only the setting of tolerances for the stopping criteria of the outer iteration and the line-search, respectively. Except in the pseudo-smooth case where a sub-sequence of quasi-limiting gradients converge to zero, the length of the search direction as well as its cosine with some generalized gradient decline monotonically like $\mathcal{O}(1/\sqrt{k})$. One might suspect that this is only a good idea in the catchment of a local minimizer, but that a reset of d might be needed when the iteration escapes from a saddle point and should start moving down a steep slope. Without reset on the Euclidean norm example (25) the convergence rate of function values and solution errors is exactly $\mathcal{O}(1/\sqrt{k})$, which we expect to be typical for sharp minimizers in general.

As mentioned in the introduction, the main thrust of this investigation is the development of an algorithmic framework that allows the provision of a software package for the solution of general semismooth optimization problems. It should avoid much of the trail and error aspect of the currently used methodology and still achieve a similar if not superior computational efficiency. Naturally, much remains to be done to this end. Most importantly, one should envision a careful implementation that utilizes abs-smooth and euc-smooth structure. Furthermore, the repeated occurence of null steps can be efficently exploited to compute a descent direction within a finite number of iterations. However, resetting the direction d, stopping criteria and a posteriori solution analysis remain significant challenges.

Acknowlegdement

The authores thank Enrico Bergmann for many inspiring discussions about the content of this paper. The research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID:390685689).

References

[1] W. Alt. *Numerische Verfahren der konvexen, nichtglatten Optimierung.* Vieweg + Teubner Verlag, 2004.

[2] M. Al-Baali. "Descent property and global convergence of the Fletcher-Reeves method with inexact line search". In: *IMA J. Numer. Anal.* 5.1 (1985), pp. 121–124. URL: `https://doi.org/10.1093/imanum/5.1.121`.

[3] A. Bagirov, N. Karmitsa, and M. Mäkelä. *Introduction to nonsmooth optimization. Theory, practice and software.* Springer, 2014.

[4] A. M. Bagirov, N. Karmitsa, and S. Taheri. *Nonsmooth Optimization Methods.* Springer, 2020.

[5] J. V. Burke et al. "Gradient Sampling Methods for Nonsmooth Optimization". In: *Numerical Nonsmooth Optimization: State of the Art Algorithms.* Ed. by A. M. Bagirov et al. Cham: Springer International Publishing, 2020, pp. 201–225. URL: `https://doi.org/10.1007/978-3-030-34910-3_6`.

[6] C. Christof, J. C. De los Reyes, and C. Meyer. “A Nonsmooth Trust-Region Method for Locally Lipschitz Functions with Application to Optimization Problems Constrained by Variational Inequalities”. In: *SIAM Journal on Optimization* 30.3 (2020), pp. 2163–2196. URL: https://doi.org/10.1137/18M1164925.

[7] F. H. Clarke. *Optimization and nonsmooth analysis.* Second. Vol. 5. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990, pp. xii+308. URL: https://doi.org/10.1137/1.9781611971309.

[8] Y.-H. Dai. “Nonlinear Conjugate Gradient Methods”. In: *Wiley Encyclopedia of Operations Research and Management Science.* American Cancer Society, 2011. URL: https://doi.org/10.1002/9780470400531.eorms0183.

[9] L. C. Evans and R. F. Gariepy. *Measure Theory and Fine Properties of Functions.* Studies in Advanced Mathematics. CRC Press, Boca Raton, FL, 1992, pp. viii+268.

[10] V. Faber et al. “Minimal residual method stronger than polynomial preconditioning”. In: *SIAM J. Matrix Anal. Appl.* 17.4 (1996), pp. 707–729. URL: https://doi.org/10.1137/S0895479895286748.

[11] S. Fiege et al. “Algorithmic differentiation for piecewise smooth functions: a case study for robust optimization”. In: *Optimization Methods and Software* (2017). available online DOI 10.1080/10556788.2017.1333613.

[12] R. Fletcher and C. M. Reeves. “Function minimization by conjugate gradients”. In: *Comput. J.* 7 (1964), pp. 149–154. URL: https://doi.org/10.1093/comjnl/7.2.149.

[13] J.-L. Goffin. “Subgradient optimization in nonsmooth optimization (including the soviet revolution).” In: *Documenta Mathematica* Extra Vol. (2012), pp. 277–290.

[14] A. A. Goldstein. “Optimization of Lipschitz continuous functions”. In: *Mathematical Programming* 13 (1977), pp. 14–22.

[15] T. Goldstein and S. Osher. “The Split Bregman Method for L1-Regularized Problems”. In: *SIAM J. Img. Sci.* 2.2 (Apr. 2009), pp. 323–343. URL: https://doi.org/10.1137/080725891.

[16] A. Griewank and A. Walther. “Beyond the Oracle: Opportunities of Piecewise Differentiation”. In: *Numerical nonsmooth optimization. State of the art algorithms.* Springer, 2020, pp. 331–361. URL: https://doi.org/10.1007/978-3-030-34910-3_10.

[17] A. Griewank and A. Walther. “Finite convergence of an active signature method to local minima of piecewise linear functions.” In: *Optimization Methods & Software* 34.5 (2019), pp. 1035–1055.

[18] A. Griewank et al. “On Lipschitz optimization based on gray-box piecewise linearization”. In: *Mathematical Programming Series A* 158.1-2 (2016), pp. 383–415.

[19] A. Griewank. “Automatic directional differentiation of nonsmooth composite functions”. In: *Recent developments in optimization (Dijon, 1994)*. Vol. 429. Lecture Notes in Econom. and Math. Systems. Springer, Berlin, 1995, pp. 155–169. URL: https://doi.org/10.1007/978-3-642-46823-0_13.

[20] A. Griewank and A. Walther. *Evaluating derivatives. Principles and techniques of algorithmic differentiation*. Second. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008, pp. xxii+438. URL: https://doi.org/10.1137/1.9780898717761.

[21] M. R. Hestenes and E. Stiefel. “Methods of conjugate gradients for solving linear systems”. In: *Journal of research of the National Bureau of Standards* 49 (1952), pp. 409–435.

[22] K. A. Khan and P. I. Barton. “Evaluating an Element of the Clarke Generalized Jacobian of a Composite Piecewise Differentiable Function”. In: *ACM Trans. Math. Softw.* 39.4 (July 2013). URL: https://doi.org/10.1145/2491491.2491493.

[23] A. Lewis and M. Overton. “Nonsmooth optimization via quasi-Newton methods”. In: *Mathematical Programming Series A* 141.1-2 (2013), pp. 135–163.

[24] A. Lewis and S. J. Wright. “A proximal method for composite minimization”. In: *Mathematical Programming* 158 (2016), pp. 501–546.

[25] R. Mifflin. “Semismooth and semiconvex functions in constrained optimization”. In: *SIAM J. Control Optim.* 15.6 (1977), pp. 959–972. URL: https://doi.org/10.1137/0315061.

[26] J. Nocedal and S. J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006, pp. xxii+664.

[27] W. d. Oliveira and C. Sagastizábal. “Bundle methods in the XXIst century: A birds’-eye view”. In: *Pesquisa Operacional* 34.3 (2014), pp. 647–670.

[28] B. T. Polyak. “Some methods of speeding up the convergence of iteration methods”. In: 1964.

[29] M. J. D. Powell. “Nonconvex minimization calculations and the conjugate gradient method”. In: *Numerical analysis (Dundee, 1983)*. Vol. 1066. Lecture Notes in Math. Springer, Berlin, 1984, pp. 122–141. URL: https://doi.org/10.1007/BFb0099521.

[30] R. Pytlak and T. Tarnawski. “On the method of shortest residuals for unconstrained optimization”. In: *J. Optim. Theory Appl.* 133.1 (2007), pp. 99–110. URL: https://doi.org/10.1007/s10957-007-9194-0.

[31] L. I. Rudin, S. Osher, and E. Fatemi. “Nonlinear total variation based noise removal algorithms”. In: *Physica D: Nonlinear Phenomena* 60.1 (1992), pp. 259–268. URL: https://www.sciencedirect.com/science/article/pii/016727899290242F.

[32] S. Scholtes. *Introduction to piecewise differentiable equations.* SpringerBriefs in Optimization. Springer, New York, 2012, pp. x+133. URL: https://doi.org/10.1007/978-1-4614-4340-7.

[33] A. Shapiro. "On concepts of directional differentiability". In: *J. Optim. Theory Appl.* 66.3 (1990), pp. 477–487. URL: https://doi.org/10.1007/BF00940933.

[34] A. Shapiro. "Sensitivity analysis of nonlinear programs and differentiability properties of metric projections". In: *SIAM J. Control Optim.* 26.3 (1988), pp. 628–645. URL: https://doi.org/10.1137/0326037.

[35] D. Sun and J. Sun. "Strong Semismoothness of the Fischer-Burmeister SDC and SOC Complementarity Functions". In: *Mathematical Programming* 103 (2005), pp. 575–581. URL: https://doi.org/10.1007/s10107-005-0577-4.

[36] A. Walther and A. Griewank. "Characterizing and testing subdifferential regularity in piecewise smooth optimization." In: *SIAM Journal on Optimization* 29.2 (2019), pp. 1473–1501. URL: https://doi.org/10.1137/17M115520X.

[37] P. Wolfe. "A method of conjugate subgradients for minimizing nondifferentiable functions". In: *Mathematical Programming Studies* 3 (1975), pp. 145–173.

[38] G. Zoutendijk. "Nonlinear programming, computational methods". In: *Integer and nonlinear programming.* 1970, pp. 37–86.