Trust-region methods without using derivatives: Worst case complexity and the non-smooth case

R. Garmanjani * D. Júdice † L. N. Vicente‡

June 9, 2016

Abstract

Trust-region methods are a broad class of methods for continuous optimization that found application in a variety of problems and contexts. In particular, they have been studied and applied for problems without using derivatives.

The analysis of trust-region derivative-free methods has focused on global convergence, and they have been proved to generate a sequence of iterates converging to stationarity independently of the starting point. Most of such an analysis is carried out in the smooth case, and, moreover, little is known about the complexity or global rate of these methods.

In this paper, we start by analyzing the worst case complexity of general trust-region derivative-free methods for smooth functions. For the non-smooth case, we propose a smoothing approach, for which we prove global convergence and bound the worst case complexity effort. For the special case of non-smooth functions that result of the composition of smooth and non-smooth/convex components, we show how to improve the existing results of the literature and make them applicable to the general methodology.

Keywords: Trust-region methods, derivative-free optimization (DFO), worst case complexity (WCC), non-smoothness, smoothing, composite functions.

1 Introduction

1.1 Trust-region methods for DFO

Trust-region methods are iterative methods for the optimization of a function in a continuous space, possibly subject to constraints. In these methods, to obtain a trial point, one typically considers the minimization of a quadratic model in a region around the current iterate and measured by a certain radius. The model serves as a local approximation of the function, in particular of its curvature (see the extensive monograph by Conn, Gould, and Toint [11] and the recent survey paper by Yuan [34]).

*CMUC, Department of Mathematics, University of Coimbra, 3001-501 Coimbra, Portugal (nima@mat.uc.pt). Support for this author was provided by FCT under the scholarship SFRH/BPD/89903/2012.

†Department of Mathematics, University of Coimbra, 3001-501 Coimbra, Portugal (judice.diogo@gmail.com). Support for this author was provided by FCT under the scholarship SFRH/BD/74401/2010.

‡CMUC, Department of Mathematics, University of Coimbra, 3001-501 Coimbra, Portugal (lnv@mat.uc.pt). Support for this author was provided by FCT under the grants PTDC/MAT/116736/2010 and PEst-C/MAT/UI0324/2011.

This paper concerns trust-region methods for unconstrained derivative-free optimization (DFO), where it is assumed that there is only access to the function values. Derivatives, if they exist, are unavailable or little reliable to be used. DFO problems are common in Engineering Optimization where the evaluation of the functions may be the output of a numerical solution. DFO has also been relatively well studied (see the book by Conn, Scheinberg, and Vicente [15]). In DFO trust-region methods, the models are frequently built by fitting a sample set using interpolation or regression, and their quality is measured by the accuracy they provide relatively to a Taylor expansion. In particular, fully linear models [13] are those as smooth and accurate as first-order Taylor ones.

Accepting the trial point as the new iterate and updating the trust-region radius depend on how much the function was reduced relatively to the model. If the current iterate is non-stationary and the model has good quality, the algorithms succeed in accepting a trial point as a new iterate in a finite number of reductions of the trust-region radius. These methods have been shown to be convergent to first-order stationary points by Conn, Scheinberg, Toint, and Vicente (in the papers [12, 14]) under the condition that fully linear models are available when necessary. The strict need of controlling geometry or considering model-improvement steps was questioned in [20], where good numerical results were reported for an interpolation-based trust-region method which ignores the geometry of the sample sets. Scheinberg and Toint [31] gave an example showing that geometry cannot be totally ignored and that some form of model improvement is necessary, at least when the size of the model gradient becomes small (a procedure known as the 'criticality step', which then ensures that the trust-region radius converges to zero).

1.2 Worst case complexity in DFO

For a long while, DFO methods have been analyzed by establishing their global convergence properties, meaning their asymptotic convergence to stationary regardless of the starting point (see [15, 25]). More recently, there has been some interest in establishing their global rates of convergence or, similarly, bounds on the number of iterations (and of function evaluations) required in the worst case to achieve a certain threshold of stationarity.

In part, such a recent effort follows a similar trend occurred for the unconstrained, derivative-based optimization of smooth functions (where the gradient exists and is Lipschitz continuous). Nesterov [27] started by showing that the gradient or steepest descent method takes at most $\mathcal{O}(\epsilon^{-2})$ ($\mathcal{O}(\epsilon^{-1})$ in the presence of convexity) iterations to drive the norm of the gradient of the objective function below ϵ. It is known that such a bound is sharp or tight (see the example of Cartis, Gould, and Toint [3]). A similar worst case complexity (WCC) bound of $\mathcal{O}(\epsilon^{-2})$ has been proved by Gratton, Sartenaer, and Toint [24] for trust-region methods. The WCC bound on the number of iterations can be reduced to $\mathcal{O}(\epsilon^{-1.5})$ for cubic overestimation methods (see Nesterov and Polyak [29] and Cartis, Gould, and Toint [5]).

In the context of DFO, most of the WCC analysis has been carried out for direct-search methods of directional type based on a sufficient decrease condition. The first WCC bound, of $\mathcal{O}(\epsilon^{-2})$, was derived by Vicente [32] for smooth functions, and later refined to $\mathcal{O}(\epsilon^{-1})$ when the function is convex by Dodangeh and Vicente [17]. Garmanjani and Vicente [21], using a smoothing approach, have shown a WCC bound of $\mathcal{O}(|\log\epsilon|\epsilon^{-3})$ in the non-smooth case. Similar WCC bounds were derived, in expectation, by Nesterov [28] for his random Gaussian smoothing approach. Cartis, Gould, and Toint [6] have derived a WCC bound of $\mathcal{O}(\epsilon^{-1.5})$

for their derivative-free adaptive cubic overestimation algorithm, but using finite differences to approximate derivatives.

1.3 The contribution of this paper

In this paper we address the WCC of trust-region methods for unconstrained DFO. Our contribution is threefold.

First we consider the smooth case and, as expected, derive a WCC bound of $\mathcal{O}(\epsilon^{-2})$ for the number of iterations and $\mathcal{O}(n^2\epsilon^{-2})$ for the number of function evaluations. There were a number of delicate issues to overcome, one of which being how to appropriately measure the effort of the criticality step to avoid worsening the power ϵ^{-2} in terms of function evaluations. It is also nontrivial to appropriately count the number of iterations that are acceptable (the function is decreased, the trial point is accepted as the new iterate, and the radius is reduced) or of model-improvement type (the iterate and the radius are maintained), under the general setting in [14].

Secondly, we address the general non-smooth case, and develop a smoothing trust-region approach in the same vein as it was first done for direct search [21] and later for sampling methods using Monte-Carlo simulation [9]. The number of iterations required to drive the smoothing parameter and the norm of the smoothing gradient below ϵ will be shown to be of $\mathcal{O}(|\log\epsilon|\epsilon^{-3})$ (for function evaluations, $\mathcal{O}(n^2|\log\epsilon|\epsilon^{-3})$). The knowledge of the contribution [21] has provided some guidance on how to obtain this result, but a lot still had to be done, from building all necessary blocks from the smooth case to assembling all components in the new context of trust regions.

The third contribution addresses the analysis of WCC of derivative-free trust-region methods for composite functions of the type $h(F)$ where h is real, non-smooth, and convex and F is vectorial and smooth (but for which derivatives are unavailable). This task was already attempted by Grapiglia, Yuan, and Yuan [23] but under a restrictive setting (relatively to the general scenario in [14]) and with sub-optimal results. Their complexity result is of the form $\mathcal{O}(|\log\epsilon|\epsilon^{-2})$, where ours will be just $\mathcal{O}(\epsilon^{-2})$. We were able to remove the factor $|\log\epsilon|$ precisely from the way we count iterations in the criticality step. Further, contrary to [23], we do not impose a reduction of the trust-region radius on model-improvement iterations. In terms of function evaluations, our bound looks like $\mathcal{O}(\ell n^2\epsilon^{-2})$, where ℓ is the number of functions components in F.

We organized our paper as follows. We start by reviewing the concept of fully linear models in Section 2. Then our three contributions are described in the following sections: Section 3 for the smooth case, Section 4 for the non-smooth case using a smoothing approach, and Section 5 for the non-smooth composite case. We provide a numerical illustration of the latter two approaches for the case $\|F\|_1$ in Section 6 and end the paper with some conclusions in Section 7.

The notation $\mathcal{O}(A)$ will mean a scalar times A, where the scalar does not depend on the iteration counter of the method under analysis (thus depending only on the problem or on algorithmic constants). The dependence of A on the dimension n of the problem (or on a Lipschitz constant) will be made explicit whenever appropriated. The notation $B(x;\Delta)$ stands for $\{y \in \mathbb{R}^n : \|y - x\| \leq \Delta\}$ and by default all norms are the Euclidean ones. Finally, $\log(\cdot)$ stands for $\ln(\cdot)$.

2 Fully linear models

Let $x_0 \in \mathbb{R}^n$ be a starting point for the trust-region methods considered in this paper. Let $F = (F_1, \ldots, F_\ell) : \mathbb{R}^n \to \mathbb{R}^\ell$ be a function for which one build models to be used in such methods. When imposing a certain smoothness on F, one needs to consider only the region where these methods generate new iterates and trial points. Given that trust-region methods impose some form of decrease on the acceptance of new iterates, such points are always confined to an initial level set $L(x_0)$. Such a level set is left undefined for the moment since it will take different forms in this paper depending on the type of problem and its smoothness. It will be of the form $\{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ if the goal is to minimize a function $f : \mathbb{R}^n \to \mathbb{R}$.

At each iteration of such methods, the function is sampled at the trial point $x_k + s_k$ and possibly at a certain number of sampling points in the ball $B(x_k; \Delta_k)$, where x_k is the current iterate and Δ_k the current trust-region radius. It might happen, however, that some of such points fall outside of the level set $L(x_0)$, and thus the set in which the function is sampled is taken as:

$$L_{enl}(x_0) \;=\; \bigcup_{x \in L(x_0)} B(x; \Delta_{max}), \tag{1}$$

where Δ_{max} is chosen such that $\Delta_{max} \geq \Delta_k$, for all $k \geq 0$. It is in $L_{enl}(x_0)$ that F is assumed smooth to later derive the convergence and complexity properties for these methods.

Assumption 2.1 *Suppose x_0 and Δ_{max} are given. Assume that F is continuously differentiable with Lipschitz continuous Jacobian (with constant L_{J_F}) in an open domain containing the set $L_{enl}(x_0)$.*

To establish global convergence to first-order stationary points (and the corresponding rates or complexity bounds), certain models of F need to be assumed as accurate as first-order Taylor models, in the sense of Point 1 of the definition below. It is further assumed that such models can be made first-order accurate or *fully linear* in a finite number of model-improvement steps. We reproduce below Definition 10.3 in [15] of fully linear models, adapting it for the case of vectorial functions, where ℓ can be greater than 1.

Definition 2.1 *Let a function $F = (F_1, \ldots, F_\ell) : \mathbb{R}^n \to \mathbb{R}^\ell$, that satisfies Assumption 2.1, be given. A set of model functions $M = \{m = (m_1, \ldots, m_\ell) : \mathbb{R}^n \to \mathbb{R}^\ell,\, m \in C^1\}$ is called a fully linear class of models if:*

1. *There exist positive constants κ_{ef} and κ_{eg} such that for any $x \in L(x_0)$ and $\Delta \in (0, \Delta_{max}]$ there exists a model function $m(x+s)$ in M, with Lipschitz continuous Jacobian, and such that*

   - *the error between the gradient of the model components and the gradient of the function components satisfies*

   $$\max_{1 \leq i \leq \ell} \|\nabla F_i(x+s) - \nabla m_i(x+s)\| \;\leq\; \kappa_{eg}\, \Delta, \quad \forall s \in B(0; \Delta), \tag{2}$$

   and

   - *the error between the model and function components satisfies*

   $$\max_{1 \leq i \leq \ell} |F_i(x+s) - m_i(x+s)| \;\leq\; \kappa_{ef}\, \Delta^2, \quad \forall s \in B(0; \Delta). \tag{3}$$

Such a model m is called fully linear on $B(x;\Delta)$.

2. *For this class M there exists an algorithm, which we will call a 'model-improvement' algorithm, that in a finite, uniformly bounded (with respect to x and Δ) number of steps can*

   - *either establish that a given model $m \in M$ is fully linear on $B(x;\Delta)$ (we will say that a certificate has been provided),*
   - *or find a model $m \in M$ that is fully linear on $B(x;\Delta)$.*

Note that when $\ell = 1$, Definition 2.1 coincides with [15, Definition 10.3]. Fully linear models are not necessarily linear, in fact they are typically quadratic in practice (see [15] for a comprehensive coverage of the topic).

3 WCC in the smooth case

This section is devoted to establishing the WCC analysis of derivative-free trust-region methods for the unconstrained minimization of smooth functions $f : \mathbb{R}^n \to \mathbb{R}$. At each iteration k of these methods, a quadratic model is form around the current iterate x_k

$$m_k(x_k + s) \;=\; f_k + g_k^\top s + \frac{1}{2} s^\top H_k s,$$

where $f_k \in \mathbb{R}$ (not necessarily equal to $f(x_k)$), $g_k \in \mathbb{R}^n$, and $H_k \in \mathbb{R}^{n\times n}$. The model is then minimized (possibly approximately) in a trust region, typically defined by a ball $B(x_k;\Delta_k)$ centered at x_k and of radius Δ_k. The difference relatively to derivative-based trust-region methods is that the models are computed based on sample values of f, and thus g_k is not necessarily the gradient of f at x_k, although it is a good approximation thereof if the model is fully linear. The matrix H_k provides an approximation to the curvature of f.

For the derivation of the WCC bounds we introduce two modifications in the presentation of the derivative-free trust-region method stated in Algorithm 4.1 in [14] (see also [15, Algorithm 10.1]).

The first modification concerns how the so-called criticality step is incorporated (see Algorithm 4.2 in [14] or the presentation in [15, Algorithm 10.2]). One knows from the counter example in [31] that such a step is indeed necessary. What the criticality step does is to improve the accuracy of the models when the model gradient g_k becomes small, ensuring that at the end of the process one has a fully linear model in a ball $B(x_k;\Delta_k)$ where Δ_k is of the order of $\|g_k\|$. In this paper, for the purpose of measuring the overall effort of the trust-region method, we consider each inner iteration of the criticality step as a regular trust-region iteration. By doing so we avoid the use of incumbent models (as done in [14]), which had to be used when the criticality step was invoked and changed the models coming from the previous iteration.

The second modification generalizes [14] by subtracting to the actual decrease $f(x_k) - f(x_k + s_k)$ a multiple of a power of the trust-region radius. The idea is that if an iteration is successful, then the actual decrease is larger than the predicted decrease plus a term of the form $c_1\Delta_k^p$, where $c_1 \geq 0$ and $p > 1$. When $c_1 = 0$ we recover the traditional scenario. When $c_1 > 0$, the additional term will allow us to derive complexity bounds dependant of p. In particular, the choice $p = 3/2$ will ask more from successful steps and lead to a worse WCC bound of $\mathcal{O}(\epsilon^{-3})$,

but such a choice will be instrumental in the analysis of complexity of the smoothing trust-region approach of Section 4.

Algorithm 3.1 Derivative-free trust-region method (for smooth functions)

Initialization: Choose an initial point x_0 and an initial trust-region radius $\Delta_0 \in (0, \Delta_{max}]$ for some $\Delta_{max} > 0$. Choose an initial model $m_0(x_0 + s)$. The constants η_0, η_1, γ, γ_{inc}, λ, and β are given and satisfy the conditions $0 \leq \eta_0 \leq \eta_1 < 1$ (with $\eta_1 \neq 0$), $\gamma \in (0, 1)$, $\gamma_{inc} > 1$, and $\lambda > \beta > 0$. Let $c_1 \geq 0$ and $p > 1$. Set $k = 0$.

Step 1 (one step of the criticality step): If $\Delta_k > \lambda \|g_k\|$, then set $x_{k+1} = x_k$. Apply the model-improvement algorithm to compute a fully linear model m_{k+1} in $B(x_{k+1}; \gamma\Delta_k)$. If the next iteration skips the criticality step (meaning $\gamma\Delta_k \leq \lambda\|g_{k+1}\|$), set $\Delta_{k+1} = \min\{\Delta_k, \max\{\gamma\Delta_k, \beta\|g_{k+1}\|\}\}$. If not, set $\Delta_{k+1} = \gamma\Delta_k$. Increment k by one and restart a new iteration in Step 1. Otherwise ($\Delta_k \leq \lambda\|g_k\|$) and go to Step 2.

Step 2 (step calculation): Compute a step s_k that sufficiently reduces the model m_k, in the sense of

$$m_k(x_k) - m_k(x_k + s_k) \;\geq\; \frac{\kappa_{fcd}}{2}\|g_k\| \min\left\{\frac{\|g_k\|}{\|H_k\|}, \Delta_k\right\} \tag{4}$$

(with $\kappa_{fcd} \in (0, 1]$), and such that $x_k + s_k \in B(x_k; \Delta_k)$.

Step 3 (acceptance of the trial point): Compute $f(x_k + s_k)$ and define

$$\rho_k \;=\; \frac{f(x_k) - f(x_k + s_k) - c_1\Delta_k^p}{m_k(x_k) - m_k(x_k + s_k)}.$$

If $\rho_k \geq \eta_1$ or if $\rho_k \geq \eta_0$ and m_k is fully linear, then $x_{k+1} = x_k + s_k$ and the model is updated to take into consideration the new iterate, resulting in a new model $m_{k+1}(x_{k+1} + s)$. Otherwise the model and the iterate remain unchanged ($m_{k+1} = m_k$ and $x_{k+1} = x_k$).

Step 4 (model improvement): If $\rho_k < \eta_1$ use a model-improvement algorithm to

- attempt to certify that m_k is fully linear on $B(x_k; \Delta_k)$,
- if such a certificate is not obtained, we say that m_k is not certifiably fully linear and make one or more suitable improvement steps.

Define $m_{k+1}(x_k + s)$ to be the improved model.

Step 5 (trust-region radius update): Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \min\{\gamma_{inc}\Delta_k, \Delta_{max}\}] & \text{if } \rho_k \geq \eta_1, \\ \{\gamma\Delta_k\} & \text{if } \rho_k < \eta_1 \text{ and } m_k \text{ is fully linear}, \\ \{\Delta_k\} & \text{if } \rho_k < \eta_1 \text{ and } m_k \\ & \text{is not certifiably fully linear.} \end{cases}$$

Increment k by one and go to Step 1.

There are essentially five types of trust-region iterations resulting from Algorithm 3.1 (critical, successful, acceptable, unsuccessful, model-improvement) but we will split the critical iterations in two types depending on whether the trust-region radius is reduced or not. Below is a description of these iterations and the symbols used to define their indices.

1. **Critical iterations ($\mathcal{C}^r$)**, taken at Step 1 and where the trust-region radius is reduced.

2. **Critical iterations ($\mathcal{C}^{nr}$)**, taken at Step 1 and where the trust-region radius is not reduced.

3. **Successful iterations ($\mathcal{S}$)**, taken at Step 3 when $\rho_k \geq \eta_1$ (the trial point is accepted and the trust-region radius is kept or increased).

4. **Acceptable iterations ($\mathcal{A}$)**, taken at Step 3 when $\rho_k \geq \eta_0$ and the model is fully linear (the trial point is accepted and the trust-region radius is decreased).

5. **Unsuccessful iterations ($\mathcal{U}$)**, taken at Step 3 when $\rho_k < \eta_0$ and m_k is fully linear (the iterate is kept and the trust-region radius is reduced).

6. **Model-improving ($\mathcal{M}$)**, taken at Step 4 when $\rho_k < \eta_1$ and m_k is not certifiably fully linear (the iterate and the trust-region radius are kept but the model is improved).

Whenever there are (more than one) consecutive model-improvement steps, we count the whole series of them as one model-improvement iteration. We know that the cost in function evaluations of such an iteration in $\mathcal{M}$ (or any iteration in $\mathcal{C}$) is of the order of n for a single function (see [15, Chapter 2]).

For analyzing the algorithm, we gather all iterations that are not successful in $\mathcal{N} = \mathcal{C} \cup \mathcal{A} \cup \mathcal{U} \cup \mathcal{M}$, where $\mathcal{C} = \mathcal{C}^r \cup \mathcal{C}^{nr}$, and all iterations where Δ_k is reduced in $\mathcal{R} = \mathcal{C}^r \cup \mathcal{A} \cup \mathcal{U}$. Due to Step 1, a reduction of the trust-region radius in the criticality step ($k \in \mathcal{C}^r$) can occur in two forms: (i) $\Delta_{k+1} = \gamma\Delta_k$ (when k is not the last iteration of a series of critical iterations or when it is and $\beta\|g_{k+1}\| \leq \gamma\Delta_k$), and in such a case we will say that $k \in \mathcal{C}^r_\gamma$. (ii) $\Delta_{k+1} = \beta\|g_{k+1}\|$ (when k is the last iteration of a series of critical iterations and $\gamma\Delta_k < \beta\|g_{k+1}\| < \Delta_k$). The trust-region radius is never increased in the criticality step, being kept constant when k is the last iteration of a series and $\beta\|g_{k+1}\| \geq \Delta_k$.

The two modifications described above do not restrict the general setting of [14]. However, a careful reader would notice that in [14] the criticality step is only applied when $\|g_k\| \leq \epsilon_c$, with $\epsilon_c > 0$. In our algorithmic presentation this would mean that a series of critical iterations is only started under the same condition. Doing this however does not affect our theory. It certainly does not have any impact on the analysis of global convergence. Selecting ϵ_c appropriately, e.g., $\epsilon_c \geq \epsilon$ when $p = 1$, where ϵ is the threshold of stationarity, would not change the analysis of WCC too. We will explain this in due course.

Given that substantial modifications in the presentation of the algorithm are made relatively to the original description in [14], it becomes necessary to ensure that the global convergence theory is still true. Part of it would have to be done anyhow for the sole purpose of analyzing the WCC.

As in the convergence of most trust-region methods, we need to assume that the objective function is bounded from below in the initial level set $L(x_0)$ and the model Hessians are uniformly bounded. The function f is assumed to satisfy Assumption 2.1 (with $f = F$, $L_{\nabla f} = L_{J_F}$, and $\ell = 1$).

Assumption 3.1 *Assume f is bounded below on $L(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$, that is there exists a constant f_{low} such that, for all $x \in L(x_0)$, $f(x) \geq f_{low}$.*

Assumption 3.2 *There exists a constant $\kappa_{bhm} > 0$ such that, for all x_k generated by the algorithm, $\|H_k\| \leq \kappa_{bhm}$.*

We will first state that the trust-region radius converges to zero. The proof is a modification of the proof of Lemma 5.5 in [14] (see also [15, Lemma 10.9]) and is left to the Appendix of the paper.

Lemma 3.1 *Let Assumptions 3.1 and 3.2 hold. Then* $\lim_{k \to +\infty} \Delta_k = 0$.

Having in mind the complexity results and the smoothing trust-region approach of Section 4, we proceed by showing that the gradient of the objective function is of the order of the trust-region radius whenever this one is reduced. The proof is left to the Appendix to let the paper flow better, but one can see there that a big part of the argument is devoted to handle the new way of counting critical iterations.

Lemma 3.2 *Let Assumptions 2.1 and 3.2 hold. If k is an iteration for which Δ_k is reduced, then*

$$\|\nabla f(x_k)\| \;\leq\; C_1 \Delta_k + C_2 \Delta_k^{p-1},$$

where

$$C_1 = \kappa_{eg} + C_0, \quad C_0 \;=\; \frac{1}{\min\left\{\beta, \frac{1}{\kappa_{bhm}}, \frac{\kappa_{fcd}(1-\eta_1)}{8\kappa_{ef}}\right\}}, \quad \textit{and} \quad C_2 = \frac{4c_1}{\kappa_{fcd}(1-\eta_1)}. \tag{5}$$

A global convergence result of the type $\liminf_{k \to +\infty} \|\nabla f(x_k)\| = 0$ follows directly from Lemma 3.2 and the asymptotic behavior of the trust-region radius.

We now derive the WCC analysis of Algorithm 3.1. We first need the following technical lemma establishing a lower bound on the trust-region radius when the size of the gradient (of the objective function) is larger than a given threshold (see the Appendix for a short proof).

Lemma 3.3 *Let Assumptions 2.1 and 3.2 hold. Let $\epsilon \in (0,1)$. Let k_0 be the first iteration where Δ_k is reduced. For every iteration $k \geq k_0$ of the algorithm, if $\|\nabla f(x_j)\| > \epsilon$ for $j = k_0, \ldots, k$, then*

$$\Delta_k \;\geq\; \gamma C_3 \epsilon^{\frac{1}{\min(p-1,1)}}, \tag{6}$$

where $C_3 = \min\left(1, (C_1 + C_2)^{-\frac{1}{\min(p-1,1)}}\right)$, with C_1 and C_2 given in (5).

We are now ready to count the number of successful iterations.

Theorem 3.1 *Let Assumptions 2.1, 3.1, and 3.2 hold. Let k_0 be the index of the first iteration where Δ_k is reduced (which must exist from Lemma 3.1). Given any $\epsilon \in (0,1)$, assume that $\|\nabla f(x_{k_0})\| > \epsilon$ and let $\bar{k}$ be the first iteration after k_0 such that $\|\nabla f(x_{\bar{k}})\| \leq \epsilon$. Then, to achieve $\|\nabla f(x_{\bar{k}})\| \leq \epsilon$, starting from k_0, Algorithm 3.1 takes at most $|\mathcal{S}(k_0, \bar{k})|$ successful iterations, where*

$$|\mathcal{S}(k_0, \bar{k})| \leq \frac{f(x_{k_0}) - f_{low}}{L} \epsilon^{-\frac{\max(p,2)}{\min(p-1,1)}}$$

with

$$L \;=\; \frac{\eta_1 \kappa_{fcd} \gamma^2 C_3^2}{2\lambda} \min\left\{\frac{1}{\kappa_{bhm}\lambda}, 1\right\} + c_1 \gamma^p C_3^p,$$

where C_3 is given in Lemma 3.3 and $\mathcal{S}(k_0, \bar{k})$ includes k_0 but excludes $\bar{k}$.

Proof. When $k \in \mathcal{S}$, using (4), $\|g_k\| \geq \Delta_k/\lambda$, and applying Lemma 3.3, we have

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta_1 \kappa_{fcd} \gamma^2 C_3^2}{2\lambda} \min\left\{\frac{1}{\kappa_{bhm}\lambda}, 1\right\} \epsilon^{\frac{2}{\min(p-1,1)}} + c_1 \gamma^p C_3^p \epsilon^{\frac{p}{\min(p-1,1)}}.$$

We then obtain by summing up all the successful iterations starting at k_0 that

$$f(x_{k_0}) - f(x_{\bar{k}}) \geq |\mathcal{S}(k_0, \bar{k})| L \epsilon^{\frac{\max(p,2)}{\min(p-1,1)}},$$

and the proof is completed. ■

It is in the counting of successful iterations that performing a series of criticality steps only when $\|g_k\| \leq \epsilon_c$ could have an impact. In fact, one would have instead $\|g_k\| \geq \min\{\epsilon_c, \Delta_k/\lambda\}$ when $k \in \mathcal{S}$. One possibility to fix the situation would be to select

$$\epsilon_c \;\geq\; \mathcal{O}(\epsilon^{\frac{1}{\min(p-1,1)}})$$

and that would only impact the constants in the result. An alternative would be to pick ϵ_c constant and consider Δ_k sufficiently small so that $\min\{\epsilon_c, \Delta_k/\lambda\} = \Delta_k/\lambda$. Such a procedure would conflict, however, with a proper WCC analysis since we would not know how many iterations would be required for Δ_k to be below $\epsilon_c\lambda$.

The next step of the analysis is to count all iterations after k_0 which are not successful.

Theorem 3.2 *Under the conditions of Theorem 3.1, to achieve $\|\nabla f(x_{\bar{k}})\| \leq \epsilon$, starting from k_0, Algorithm 3.1 takes at most $|\mathcal{N}(k_0, \bar{k})|$ other (not successful) iterations, where*

$$|\mathcal{N}(k_0, \bar{k})| \;\leq\; (3 + 4L_1)|\mathcal{S}(k_0, \bar{k})| + 4\left(L_2 - \log_\gamma(e)\epsilon^{-\frac{1}{\min(p-1,1)}}\right),$$

where C_3 is given in Lemma 3.3, $L_1 = -\log_\gamma(\gamma_{inc})$, and $L_2 = \log_\gamma\left(\frac{\gamma^2 C_3 e}{\Delta_{k_0}}\right)$.

Proof. For iterations k in $\mathcal{R}_\gamma = \mathcal{C}_\gamma^r \cup \mathcal{A} \cup \mathcal{U}$, $\Delta_{k+1} = \gamma\Delta_k$. For successful iterations $k \in \mathcal{S}$, $\Delta_{k+1} \leq \gamma_{inc}\Delta_k$. For the others ($k \in \mathcal{C}_\gamma^{nr} \cup \mathcal{M}$, where $\mathcal{C}_\gamma^{nr} = \mathcal{C} \setminus \mathcal{C}_\gamma^r$), $\Delta_{k+1} \leq \Delta_k$. Thus, we obtain by induction

$$\Delta_{\bar{k}} \;\leq\; \Delta_{k_0} \gamma_{inc}^{|\mathcal{S}(k_0,\bar{k})|} \gamma^{|\mathcal{R}_\gamma(k_0,\bar{k})|}.$$

As $\log(\gamma) < 0$, one can then write

$$|\mathcal{R}_\gamma(k_0, \bar{k})| \;\leq\; -\frac{\log(\gamma_{inc})}{\log(\gamma)}|\mathcal{S}(k_0, \bar{k})| - \frac{\log(\Delta_{k_0})}{\log(\gamma)} + \frac{\log(\Delta_{\bar{k}})}{\log(\gamma)}.$$

Lemma 3.3 guarantees (6) for $j = k_0, \ldots, \bar{k} - 1$. As $\Delta_{\bar{k}} \geq \gamma\Delta_{\bar{k}-1}$ and again because $\log(\gamma) < 0$, we have that

$$\frac{\log(\Delta_{\bar{k}})}{\log(\gamma)} \;\leq\; \frac{\log(\gamma^2 C_3)}{\log(\gamma)} + \frac{\log(\epsilon^{\frac{1}{\min(p-1,1)}})}{\log(\gamma)}.$$

Combining the last two inequalities, one obtains

$$|\mathcal{R}_\gamma(k_0, \bar{k})| \;\leq\; L_1|\mathcal{S}(k_0, \bar{k})| + \log_\gamma\left(\frac{\gamma^2 C_3}{\Delta_{k_0}}\right) - \frac{\log(\epsilon^{-\frac{1}{\min(p-1,1)}})}{\log(\gamma)}.$$

Now, using $\log(x) \leq x - 1$ for $x > 1$, we reach

$$|\mathcal{R}_\gamma(k_0, \bar{k})| \;\leq\; L_1 |\mathcal{S}(k_0, \bar{k})| + L_2 - \log_\gamma(e)\epsilon^{-\frac{1}{\min(p-1,1)}}. \tag{7}$$

It remains to count the iterations that are in $\mathcal{C}_\gamma^{nr}$ and in $\mathcal{M}$. After an iteration in $\mathcal{C}_\gamma^{nr}$ (a last critical iteration in a series of them), the model is fully linear, and thus the next iteration is either successful, acceptable, or unsuccessful, giving

$$|\mathcal{C}_\gamma^{nr}| \;\leq\; |\mathcal{S}| + |\mathcal{A}| + |\mathcal{U}| \;\leq\; |\mathcal{S}| + |\mathcal{R}_\gamma|.$$

After an iteration in $\mathcal{M}$, the next one is of one of the other types, and thus

$$|\mathcal{M}| \;\leq\; |\mathcal{S}| + |\mathcal{R}| + |\mathcal{C}_\gamma^{nr}| \;\leq\; 2(|\mathcal{S}| + |\mathcal{R}_\gamma|).$$

Thus,

$$|\mathcal{N}| \;=\; |\mathcal{R}_\gamma \cup \mathcal{C}_\gamma^{nr} \cup \mathcal{M}| \;\leq\; |\mathcal{R}_\gamma| + |\mathcal{C}_\gamma^{nr}| + |\mathcal{M}| \;\leq\; 3|\mathcal{S}| + 4|\mathcal{R}_\gamma|,$$

which combined with (7) completes the proof. ■

The two last theorems show that the number of iterations, after the first iteration k_0 where the trust-region radius is reduced, that are needed to drive the norm of the gradient below ϵ is

$$\mathcal{O}\left(\epsilon^{-\frac{\max(p,2)}{\min(p-1,1)}}\right).$$

It can be easily shown that k_0 is also bounded by such a quantity. From what we have seen in the proof of Theorem 3.2, since there are no iterations in $\mathcal{R}_\gamma$ until k_0, one has $k_0 \leq 4|\mathcal{S}(0, k_0)|$. To count the number of successful iterations up to $k_0 - 1$, we write, as in the proof of Theorem 3.1, for such iterations k,

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta_1 \kappa_{fcd}}{2\lambda} \min\left\{\frac{1}{\kappa_{bhm}\lambda}, 1\right\} \Delta_k^2 + c_1 \Delta_k^p.$$

Summing up all these iterations up to k_0, and considering $\Delta_k \geq \Delta_0$ and $\epsilon < 1$, we obtain

$$k_0 \;\leq\; 4|\mathcal{S}(0, k_0)| \;\leq\; 4\frac{f(x_0) - f(x_{k_0})}{\min\{\Delta_0^2, \Delta_0^p\} L_0} \;\leq\; 4\frac{f(x_0) - f(x_{k_0})}{\min\{\Delta_0^2, \Delta_0^p\} L_0} \epsilon^{-\frac{\max(p,2)}{\min(p-1,1)}}, \tag{8}$$

with

$$L_0 \;=\; \frac{\eta_1 \kappa_{fcd}}{2\lambda} \min\left\{\frac{1}{\kappa_{bhm}\lambda}, 1\right\} + c_1.$$

To state our final complexity result, one needs to make explicit the dependence of the constants appearing so far in terms of the problem dimension n and the Lipschitz constant of the gradient. It is known that the constants κ_{ef} and κ_{eg} in the definition of fully linear models can meet the following assumption (see, e.g., [15, Chapter 2]).

Assumption 3.3 *The constants κ_{ef} and κ_{eg} in the definition of fully linear models satisfy $\kappa_{ef} = \mathcal{O}(\sqrt{n} L_{\nabla f})$ and $\kappa_{eg} = \mathcal{O}(\sqrt{n} L_{\nabla f})$, where n is the problem dimension and $L_{\nabla f}$ is the Lipschitz constant of the gradient of the objective function f.*

Theorems 3.1 and 3.2 and the bound on k_0 given above, together with Assumption 3.3, lead to the following result

Theorem 3.3 *Let Assumptions 2.1, 3.1, 3.2, and 3.3 hold. To drive the norm of the gradient below $\epsilon \in (0,1)$, Algorithm 3.1 takes at most*

$$\mathcal{O}\left((L_{\nabla f}\sqrt{n})^{\frac{\max(p,2)}{\min(p-1,1)}} \epsilon^{-\frac{\max(p,2)}{\min(p-1,1)}} \right)$$

iterations. When $p = 2$, this number is of $\mathcal{O}(L_{\nabla f}^2 n \epsilon^{-2})$.

Proof. It suffices to observe that for the constant L appearing in Theorem 3.1 we have

$$\frac{1}{L} \;=\; \mathcal{O}\left(C_3^{-\max(p,2)}\right) \;=\; \mathcal{O}\left((C_1 + C_2)^{\frac{\max(p,2)}{\min(p-1,1)}} \right) = \; \mathcal{O}\left(\kappa^{\frac{\max(p,2)}{\min(p-1,1)}} \right),$$

with $\kappa = \max\{\kappa_{ef}, \kappa_{eg}\}$ and then to apply Assumption 3.3. ■

Algorithm 3.1 takes at most $\mathcal{O}(n)$ function evaluations at critical and model-improving iterations and only one function evaluation at all other iterations. It is then possible to measure the worst case effort also in terms of function evaluations.

Corollary 3.1 *Let Assumptions 2.1, 3.1, 3.2, and 3.3 hold. To drive the norm of the gradient below $\epsilon \in (0,1)$, Algorithm 3.1 takes at most*

$$\mathcal{O}\left(n(L_{\nabla f}\sqrt{n})^{\frac{\max(p,2)}{\min(p-1,1)}} \epsilon^{-\frac{\max(p,2)}{\min(p-1,1)}} \right)$$

function evaluations. When $p = 2$, this number is of $\mathcal{O}(L_{\nabla f}^2 n^2 \epsilon^{-2})$.

4 Smoothing trust-region methods

In this section we consider the unconstrained minimization of functions $f : \mathbb{R}^n \to \mathbb{R}$ that are locally Lipschitz continuous, but not necessarily differentiable or convex.

4.1 Smoothing functions

Given our objective function f we will assume, however, the existence and knowledge of a smoothing function (see [7, 35]):

Definition 4.1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function. We call $\tilde{f} : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}$ a smoothing function of f if, for any $\mu > 0$, $\tilde{f}(\cdot, \mu)$ is continuously differentiable in $\mathbb{R}^n$ and, for any $x \in \mathbb{R}^n$,*

$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) \;=\; f(x).$$

Under reasonable assumptions, the smoothing trust-region methods derived in this section will generate a sequence of points and a sequence of smoothing parameters (converging to zero) for which the gradient of the smoothing function tends to zero. In other words, we will show that

any limit point x_* of that sequence of points is a stationary point of the smoothing function $\tilde{f}$, in the sense that $0 \in G_{\tilde{f}}(x_*)$, with

$$G_{\tilde{f}}(x_*) \;=\; \{v : \exists N \in \mathcal{N}_\infty, (x,\mu) \underset{N}{\longrightarrow} (x_*, 0) \;\text{ with }\; \nabla \tilde{f}(x,\mu) \underset{N}{\longrightarrow} v\},$$

where $\mathcal{N}_\infty$ represents the set of infinite sequences. It is known that for certain types of objective functions and corresponding smoothing functions, $\operatorname{co} G_{\tilde{f}}(x_*) = \partial f(x_*)$, where $\partial f(x_*)$ denotes the Clarke subdifferential of f at x_*, a result that has been called gradient consistency [10]. One way to guarantee gradient consistency is subdifferential regularity of f at x_* (see [30, Theorem 9.67]). Gradient consistency has also been studied in the papers [2, 8]. Thus, in those cases, the smoothing trust-region methods are capable of generating a sequence of iterates converging to Clarke stationary points.

The method developed in this section could be used for minimizing composite functions of the type $f = g + h(F)$, where $h : \mathbb{R}^\ell \to \mathbb{R}$ is non-smooth with a known smoothing function and $g : \mathbb{R}^n \to \mathbb{R}$ and $F : \mathbb{R}^n \to \mathbb{R}^\ell$ are assumed smooth (continuously differentiable). The functions g and F can be a black box or a zero-order oracle, in the sense that one does not access to derivative information, only function values can be evaluated.

4.2 The algorithm

Following what has been done in [21] for direct search, we introduce a smoothing trust-region algorithm for the unconstrained minimization of a locally Lipschitz continuous objective function f for which a smoothing function $\tilde{f}$ is known. The idea is simple and consists of the application of Algorithm 3.1 to the smoothing function for decreasing values of the smoothing parameter μ. Each outer or main iteration (Algorithm 3.1 applied to $\tilde{f}$ for a fixed value of μ) is stopped when the trust-region radius becomes smaller than a function $r(\mu)$ of the smoothing parameter.

Algorithm 4.1 (Smoothing trust-region method)

Initialization

Choose x_{-1} with $f(x_{-1}) < +\infty$, $\Delta_0 > 0$, $\mu_0 > 0$, and $\sigma \in (0,1)$.

For $k = 0, 1, 2, \ldots$

1. **Trust-region method for a fixed smoothing parameter:** Apply Algorithm 3.1 to $\tilde{f}(\cdot, \mu_k)$ (starting from $y_{0,k} = x_{k-1}$) generating points $y_{0,k}, \ldots, y_{j_k,k}$ until $\Delta_{j_k+1,k} < r(\mu_k)$.
2. **Update of the smoothing parameter:** Set $x_k = y_{j_k,k}$ and decrease the smoothing parameter: $\mu_{k+1} = \sigma \mu_k$.

As we will see next, each outer iteration is well defined (in the sense of stopping in a finite number of inner iterations) and, moreover, Algorithm 4.1 will stop under a criterion of the form $\mu_k \leq \mu_{tol}$, where $\mu_{tol} \in (0, \mu_0)$.

4.3 Global convergence

We will analyze the global convergence of the smoothing trust-region method (Algorithm 4.1) under the following assumptions, which are the natural counterparts, for the smoothing function, of the ones assumed in the smooth case of Section 3.

Assumption 4.1 *For all k: $\tilde{f}(\cdot, \mu_k)$ has a Lipschitz continuous gradient with constant $L_{\nabla \tilde{f}}(\mu_k)$ on an open set containing $L_{enl}(y_{0,k})$, see (1), with $L(y_{0,k}) = \{y \in \mathbb{R}^n : \tilde{f}(y, \mu_k) \leq \tilde{f}(y_{0,k}, \mu_k)\}$.*

Assumption 4.2 *For all k: the functions $\tilde{f}(\cdot, \mu_k)$ are bounded below in $L(y_{0,k})$.*

Each inner iteration of Algorithm 4.1 consists of one iteration of Algorithm 3.1 using a quadratic model now written as

$$\tilde{m}_{j,k}(y_{j,k} + s, \mu_k) \;=\; \tilde{f}_{j,k} + \tilde{g}_{j,k}^{\top} s + \frac{1}{2} s^{\top} \tilde{H}_{j,k} s.$$

As in Section 3, we will require all these model Hessians to be uniformly bounded.

Assumption 4.3 *There exists a constant $\tilde{\kappa}_{bhm} > 0$ such that, for all j, k, $\|\tilde{H}_{j,k}\| \leq \tilde{\kappa}_{bhm}$.*

One can immediately deduce that the smoothing parameter converges to zero.

Theorem 4.1 *Let Assumptions 4.2 and 4.3 hold. Then $\lim_{k \to +\infty} \mu_k = 0$.*

Proof. For each k, one knows, from Lemma 3.1, that $\lim_{j \to +\infty} \Delta_{j,k} = 0$. Thus, one always reaches the stopping criterion for every k and μ_k is reduced an infinite number of times, which completes the proof. ■

The above result triggers the following one. Note that $r(\mu)$ is part of the algorithmic design and can be chosen in whatever most appropriate way.

Theorem 4.2 *Let Assumptions 4.2 and 4.3 hold. If $\lim_{\mu \downarrow 0} r(\mu) = 0$, then $\lim_{k \to +\infty} \Delta_{j_k,k} = 0$.*

Proof. The proof results from Theorem 4.1 and the fact that $r(\mu_k) \geq \Delta_{j_k+1,k} = \gamma \Delta_{j_k,k}$. ■

Global convergence of Algorithm 4.1 requires that $r(\mu)$ goes to zero faster than the way that the Lipschitz constant $L_{\nabla \tilde{f}}(\mu)$ of the gradient of the smoothing function goes to infinity (see the theorem below). Later we will see that the optimal complexity bound asks for a Lipschitz constant $L_{\nabla \tilde{f}}(\mu)$ that does not go to infinity faster than $1/\mu$, in other words that $L_{\nabla \tilde{f}}(\mu) = \mathcal{O}(1/\mu)$. There are smoothing functions satisfying this property as well as gradient consistency, such as the smoothing function for the absolute value defined by Chen and Zhou [10] and composite functions of the type $\|F\|_1$ where F is smooth [21].

Theorem 4.3 *Consider the application of Algorithm 4.1 and suppose that $\tilde{f}$ is a smoothing function for f. Let Assumptions 4.1, 4.2, and 4.3 hold. Under these conditions, if $\lim_{\mu \downarrow 0} r(\mu) = 0$ and $\lim_{\mu \downarrow 0} L_{\nabla \tilde{f}}(\mu) r(\mu) = 0$, then*

$$\lim_{k \to +\infty} \|\nabla \tilde{f}(x_k, \mu_k)\| \;=\; 0 \tag{9}$$

and any accumulation point x_* of $\{x_k\}$ is a stationary point associated with the smoothing function $\tilde{f}$.

Proof. For each k, $x_k = y_{j_k,k}$, where j_k is an iteration such that the trust-region radius is reduced. Thus, in view of Lemma 3.2, we have

$$\|\nabla \tilde{f}(x_k, \mu_k)\| \;\leq\; C_1(\mu_k)\Delta_{j_k,k} + C_2\Delta_{j_k,k}^{p-1},$$

where now $C_1 = C_1(\mu_k)$ depends on μ_k through the dependence of $L_{\nabla \tilde{f}}(\mu_k)$. Since $C_1(\mu_k) = \mathcal{O}(\tilde{\kappa}_{eg}) = \mathcal{O}(L_{\nabla \tilde{f}}(\mu_k))$, where $\tilde{\kappa}_{eg}$ is the constant in the error bound (2) for the gradient of the model of the smoothing function $\tilde{f}$, and $r(\mu_k) \geq \Delta_{j_k+1,k} = \gamma\Delta_{j_k,k}$, one obtains

$$\|\nabla \tilde{f}(x_k, \mu_k)\| \;\leq\; \mathcal{O}(L_{\nabla \tilde{f}}(\mu_k))r(\mu_k) + C_2\Delta_{j_k,k}^{p-1}.$$

Then, due to Theorems 4.1 and 4.2, we obtain (9) and the proof is completed. ∎

If one considers a smoothing function $\tilde{f}$ for which $L_{\nabla \tilde{f}}(\mu) = \mathcal{O}(1/\mu)$, it suffices to choose $r(\mu) = \mu^q$, with $q > 1$, to successfully apply Theorem 4.3.

As a consequence of the above result, when the smoothing function of f satisfies the above mentioned gradient consistency property at an accumulation point x_*, one knows that x_* is a Clarke stationary point of the function f (since $0 \in G_{\tilde{f}}(x_*) \subseteq \operatorname{co} G_{\tilde{f}}(x_*) = \partial f(x_*)$).

4.4 Worst case complexity

We also follow here the same steps as in [21] and start by first counting the number of inner iterations of Algorithm 4.1 to drive the smoothing parameter below a given threshold.

Theorem 4.4 *Consider the application of Algorithm 4.1 using the term $c_1\Delta^p$ when calling Algorithm 3.1 and $r(t) = c_2 t^q$, with $p, q > 1$ and $c_1, c_2 > 0$. Suppose that $\tilde{f}$ is a smoothing function for f. Let Assumptions 4.1, 4.2, and 4.3 hold.*

Given any $\xi \in (0,1)$ such that $\xi < \mu_0$, let $\bar{k}$ be the first outer iteration such that $\mu_{\bar{k}+1} \leq \xi$. Under these assumptions, Algorithm 4.1 takes at most $\mathcal{O}\left(|\log(\xi)|\xi^{-pq}\right)$ inner iterations to reduce the smoothing parameter below ξ, i.e., to have $\mu_{\bar{k}+1} < \xi$.

Proof. First let us consider each inner loop of Algorithm 4.1 where a trust-region method is applied for a fixed $\mu_k > \xi$. This loop is repeated until there is an iteration (j_k, k) for which the trust-region radius is reduced and $\Delta_{j_k+1,k} < r(\mu_k) = c_2\mu_k^q$.

For each k, the number of inner iterations needed to reach the first iteration $(j_{0,k}, k)$ where the trust-region radius is reduced is of the $\mathcal{O}(1)$ (see (8)).

One has, for a successful iteration (j,k), since $\Delta_{j,k} \geq c_2\mu_k^q$, that

$$\tilde{f}(y_{j,k}, \mu_k) - \tilde{f}(y_{j+1,k}, \mu_k) \;\geq\; \eta_1\frac{\tilde{\kappa}_{fcd}}{2}\|g_{j,k}\| \min\left\{\frac{\|g_{j,k}\|}{\tilde{\kappa}_{bhm}}, \Delta_{j,k}\right\} + c_1\Delta_{j,k}^p \;\geq\; c_1 c_2^p \mu_k^{pq}.$$

The number of inner successful iterations $|\mathcal{S}_k(j_{0,k}, j_k)|$ from $(j_{0,k}, k)$ until (j_k, k) is then bounded by

$$|\mathcal{S}_k(j_{0,k}, j_k)| \;\leq\; \frac{\tilde{f}(y_{j_{0,k},k}, \mu_k) - \tilde{f}_{low,k}}{c_1 c_2^p}\frac{1}{\mu_k^{pq}}.$$

Similar to the first part of the proof of Theorem 3.2, the number of the other inner iterations is bounded as follows (remember that $0 < \gamma < 1$)

$$|\mathcal{R}_k(j_{0,k}, j_k)| \;\leq\; L_1|\mathcal{S}_k(j_{0,k}, j_k)| - \log_\gamma(\Delta_{j_{0,k},k}) + \log_\gamma(\Delta_{j_k,k}).$$

The initial trust-region radii $\Delta_{j_{0,k},k}$ are considered constants. To bound the third term, recall that $\Delta_{j_k,k} \geq r(\mu_k) = c_2\mu_k^q > c_2\xi^q$, and thus, since $p > 1$, $\log_\gamma(\Delta_{j_k,k}) = \mathcal{O}(\xi^{-pq})$. We conclude that the maximum number of iterations needed in each inner loop minimization is $\mathcal{O}(\xi^{-pq})$.

Finally, let us count the number of outer loops. From the updating scheme of the smoothing parameter, one has $\mu_{k+1} \leq \sigma^k\mu_0$. Thus, the number of outer iterations required to reach $\mu_{\bar{k}+1} < \xi$ satisfies $\bar{k} \geq [\log(\xi) - \log(\mu_0)]/\log(\sigma)$, and the proof is completed. ■

There are situations where the Lipschitz constant of the gradient of the smoothing function is of the order of $1/\mu$: see [10] for the absolute value $|\cdot|$, [21] for the composite function $\|F\|_1$ with F smooth, and [28] for smoothing functions using Gaussian densities. Under such an assumption on $L_{\nabla\tilde{f}}(\mu)$ it is possible to bound the gradient of $\tilde{f}$ at the end of the last outer loop.

Theorem 4.5 *Let all the assumptions of Theorem 4.4 hold and assume also that $L_{\nabla\tilde{f}}(\mu_k) = \mathcal{O}(1/\mu_k)$. Suppose also that the constant $\tilde{\kappa} = \max\{\tilde{\kappa}_{ef}, \tilde{\kappa}_{eg}\}$ in the bounds of the fully linear models of $\tilde{f}$ satisfies Assumption 3.3.*

Given any $\xi \in (0,1)$ such that $\xi < \mu_0$, let $\bar{k}$ be the first iteration such that $\mu_{\bar{k}+1} \leq \xi$. Under these conditions, one has

$$\|\nabla\tilde{f}(x_{\bar{k}}, \mu_{\bar{k}})\| \;=\; \mathcal{O}\left(\sqrt{n}\xi^{q-1} + \xi^{(p-1)q}\right).$$

Proof. From Lemma 3.2 and $\Delta_{j_k,k} = \Delta_{j_k+1,k}/\gamma < (c_2/\gamma)\mu_k^q$, one has

$$\|\nabla\tilde{f}(x_{\bar{k}}, \mu_{\bar{k}})\| \;\leq\; C_1\Delta_{j_{\bar{k}}} + C_2\Delta_{j_{\bar{k}}}^{p-1} \;\leq\; C_1(c_2/\gamma)\mu_{\bar{k}}^q + C_2(c_2/\gamma)^{p-1}\mu_{\bar{k}}^{(p-1)q}.$$

The proof is completed by noting that $C_1 = \mathcal{O}(\tilde{\kappa}) = \mathcal{O}(\sqrt{n}L_{\nabla\tilde{f}}) = \mathcal{O}(\sqrt{n}/\mu)$ and that, from $\mu_{\bar{k}+1} = \sigma\mu_{\bar{k}}$, one has $\mu_{\bar{k}} \leq \xi/\sigma$. ■

This result suggests that $p = 3/2$ and $q = 2$ are the optimal choices in the sense that $\|\nabla\tilde{f}(x_{\bar{k}}, \mu_{\bar{k}})\|$ becomes $\mathcal{O}(\sqrt{n}\xi)$. We are thus finally ready to state a WCC bound for driving both the norm of the smoothing gradient and the smoothing parameter below a common threshold.

Corollary 4.1 *Under the assumptions of Theorem 4.5 and when $q = 2$ and $p = \frac{3}{2}$, Algorithm 4.1 takes at most $\mathcal{O}\left(|\log(\xi)|\xi^{-3}\right)$ iterations (and at most $\mathcal{O}\left(n|\log(\xi)|\xi^{-3}\right)$ function evaluations) to reduce the smoothing parameter below $\xi \in (0,1)$, ending such process with*

$$\|\nabla\tilde{f}(x_{\bar{k}}, \mu_{\bar{k}})\| = \mathcal{O}(\sqrt{n}\xi). \tag{10}$$

Equivalently, the number of iterations needed to reach $\|\nabla\tilde{f}(x_{\bar{k}}, \mu_{\bar{k}})\| \leq \epsilon$ and $\mu_{\bar{k}} \leq \xi = \epsilon/(\sqrt{n}C)$, where $C > 0$ is the constant that multiplies $\sqrt{n}\xi$ in the right hand side of (10), is

$$\mathcal{O}\left(n^{\frac{3}{2}}[|\log(\epsilon)| + \log(n)]\epsilon^{-3}\right),$$

leading to the following overall WCC bound in terms of the number of function evaluations

$$\mathcal{O}\left(n^{\frac{5}{2}}[|\log(\epsilon)| + \log(n)]\epsilon^{-3}\right).$$

5 Trust-region methods for composite functions

In this section we consider the unconstrained minimization of composite functions of the type $f = h(F)$, where $h : \mathbb{R}^\ell \to \mathbb{R}$ is a convex, possibly non-smooth function that is at least globally Lipschitz continuous (with constant $L_h > 0$). The vectorial function $F : \mathbb{R}^n \to \mathbb{R}^\ell$ is assumed smooth (continuously differentiable) but it is considered that only function values can be computed, not derivatives. The setting can be easily extended to $f = g + h(F)$ as long as $g : \mathbb{R}^n \to \mathbb{R}$ is smooth and one can build convex and fully linear models of it.

Given $x \in \mathbb{R}^n$ and $\Delta > 0$, if the Jacobian $J(x)$ of F was known, we could consider the trust-region subproblem $\min_{\|s\| \leq \Delta} l(x, s)$, where $l(x, s)$ is the following approximation of f around x (composition of h with a linear approximation of F):

$$l(x, s) \;=\; h(F(x) + J(x)s).$$

The decrease predicted by the step would then be

$$\Psi(x, \Delta) \;=\; l(x, 0) - \min_{\|s\| \leq \Delta} l(x, s).$$

$\Psi(x, 1)$ was used in [4] as a criticality measure for f. In fact, x_* is a critical point of f if and only if $\Psi(x_*, 1) = 0$ (and $\Psi(x, 1)$ is non-negative and continuous for all x), see [33, Lemma 2.1].

In this paper, since we assume that the Jacobian of F is not available, we replace $l(x, s)$ by a composite model of the form $l^m(x, s) = h(m(x + s))$, where $m(x + s) = F(x) + J^m(x)s$ is a fully linear model of F (in the sense of Definition 2.1). One possibility to compute such a model is to set the lines of the matrix $J^m(x)$ to the transposes of the simplex gradients of the components of F at x (see [15, Chapter 2]). The decrease predicted by the solution of the trust-region subproblem $\min_{\|s\| \leq \Delta} l^m(x, s)$ is then

$$\Psi^m(x, \Delta) \;=\; l^m(x, 0) - \min_{\|s\| \leq \Delta} l^m(x, s),$$

and $\Psi^m(x, 1)$ is our model of criticality measure. In practice, and when h has a piecewise linear structure such as the one given by the ℓ_1 or ℓ_∞ norms, the model $m(x + s)$ will be considered linear to render easy the solution of the trust-region subproblem.

In the following we will show that the difference between the true and the model criticality measures is of the order of the trust-region radius. This result was proved originally in [23, Theorem 1] assuming linearity of the model $m(x+s)$ in s, but it can be made simpler as we show below if we only use the fully linearity of the models. Let $t \in B(0; \Delta)$, $s_t = \operatorname{argmin}_{\|s\| \leq 1} l(x+t, s)$, and $s_t^m = \operatorname{argmin}_{\|s\| \leq 1} l^m(x + t, s)$. Consider first the case $\Psi^m(x + t, 1) \leq \Psi(x + t, 1)$. Since $l^m(x + t, s_t^m) \leq l^m(x + t, s_t)$, using (3),

$$\begin{aligned}
\Psi(x + t, 1) - \Psi^m(x + t, 1) \;&\leq\; l(x + t, 0) - l(x + t, s_t) - [l^m(x + t, 0) - l^m(x + t, s_t)] \\
&=\; h(F(x + t) + J^m(x + t)s_t) - h(F(x + t) + J(x + t)s_t) \\
&\leq\; L_h \|J(x + t) - J^m(x + t)\| \|s_t\| \;\leq\; (L_h \kappa_{eg})\, \Delta.
\end{aligned}$$

In the case $\Psi(x+t, 1) \leq \Psi^m(x+t, 1)$, it can be proved similarly that $\Psi^m(x+t, 1) - \Psi(x+t, 1) \leq (L_h \kappa_{eg})\Delta$. Therefore, we have

$$|\Psi(x + t, 1) - \Psi^m(x + t, 1)| \;\leq\; \kappa_\Psi \Delta, \quad \forall t \in B(0; \Delta), \quad \text{with} \quad \kappa_\Psi \;=\; L_h \kappa_{eg}. \tag{11}$$

5.1 The algorithm

A derivative-free trust region algorithm for composite functions can be stated in the same vein as it was done in Algorithm 3.1 for smooth functions. The differences lie uniquely in the definition of the criticality measure, in the trust-region subproblem, in the definition of the predicted decrease, and in the fact that m models F in $f = h(F)$ (instead of modeling f directly as in Algorithm 3.1). There is no need now to consider the term $c_1\Delta_k^p$ in ρ_k, as its inclusion in Algorithm 3.1 was primarily done for deriving the complexity bounds for the smoothing trust-region approach of Section 4.

Algorithm 5.1 Derivative-free trust-region method (for composite functions)

Initialization: Same as in Algorithm 3.1 but setting $c_1 = 0$.

Step 1 (criticality step): Same as in Algorithm 3.1 but with $\|g_k\|$ replaced by $\Psi^m(x_k, 1)$.

Step 2 (step calculation): Compute the step s_k by solving

$$\min_{\|s\| \leq \Delta_k} \; l^m(x_k, s).$$

Step 3 (acceptance of the trial point): Same as in Algorithm 3.1 with $m_k(x_k) - m_k(x_k + s_k)$ replaced by $\Psi^m(x_k, \Delta_k)$.

Step 4 (model improvement): Same as in Algorithm 3.1.

Step 5 (trust-region radius update): Same as in Algorithm 3.1.

Similar to Algorithm 3.1, there are six types of iterations and we will use the same notation as in Section 3. For the rest of the current section, we use Ψ_k and Ψ_k^m instead of $\Psi(x_k, 1)$ and $\Psi^m(x_k, 1)$, respectively.

5.2 Global convergence

As we said before we will require h to satisfy the following assumption.

Assumption 5.1 *The function $h : \mathbb{R}^\ell \to \mathbb{R}$ is convex, globally Lipschitz continuous, with Lipschitz constant $L_h > 0$, and bounded from below.*

The following lemma and its proof are an adaptation of Lemma 2.1 in [4].

Lemma 5.1 *Let Assumption 5.1 hold. Then $\Psi^m(x_k, \Delta_k) \geq \min\{\Delta_k, 1\}\Psi_k^m$.*

Proof. When $\Delta_k \geq 1$, from $\min_{\|s\| \leq 1} l^m(x_k, s) \geq \min_{\|s\| \leq \Delta} l^m(x_k, s)$, we have $\Psi^m(x_k, \Delta_k) \geq \Psi_k^m$. When $\Delta_k < 1$, consider $s_k^* = \operatorname{argmin}_{\|s\| \leq 1} l^m(x_k, s)$. Then,

$$\Psi^m(x_k, \Delta_k) \;\geq\; l^m(x_k, 0) - l^m(x_k, \Delta_k s_k^*) \;\geq\; \Delta_k[l^m(x_k, 0) - l^m(x_k, s_k^*)] \;=\; \Delta_k \Psi_k^m,$$

where the first inequality holds due to $l^m(x_k, s_k) \leq l^m(x_k, \Delta_k s_k^*)$ and the second inequality holds due to the convexity of l^m. ■

In our derivation of the WCC bounds we need to make sure that there exists at least one iteration for which the corresponding trust-region radius is reduced. This is guaranteed by the following lemma.

Lemma 5.2 *Let Assumption 5.1 hold. Then* $\lim_{k\to+\infty} \Delta_k = 0$.

Proof. The only differences from the proof of Lemma 3.1 lie in the use of the predicted decrease. Now, when $k \in \mathcal{S}$, we have $f(x_k)-f(x_{k+1}) \geq \eta_1 \Psi^m(x_k, \Delta_k)$. Then by using Lemma 5.1 and $\Psi_k^m \geq \Delta_k/\lambda$ (since the iteration is not in $\mathcal{C}$)

$$f(x_k) - f(x_{k+1}) \;\geq\; \eta_1 \min\{\Delta_k, 1\}\lambda^{-1}\Delta_k.$$

■

In the following lemma, which can be seen as a combination of Lemma 3.2 and Lemma 2.2 in [4], we bound the criticality measure by a constant multiple of the trust-region radius.

Lemma 5.3 *Let Assumptions 2.1 and 5.1 hold. If k is an iteration for which Δ_k is reduced, then*

$$\Delta_k \;\geq\; \min\left\{ \frac{1}{\kappa_\Psi \lambda + 1} \min\{\sqrt{C_4\Psi_k}, C_4\Psi_k\}, \frac{1}{\kappa_\Psi + 1/\beta}\Psi_k \right\},$$

where $C_4 = \frac{1-\eta_1}{2L_h\kappa_{ef}}$, and κ_Ψ comes from (11).

Proof. Let us suppose first that $k \in \mathcal{A}\cup\mathcal{U}$. To later arrive at a contradiction, suppose that

$$\Delta_k \;<\; \min\{\sqrt{C_4\Psi_k^m}, C_4\Psi_k^m\}. \tag{12}$$

Using (3) and Lemma 5.1, we have

$$|\rho_k - 1| \;=\; \frac{|h(F(x_k)) - h(m(x_k)) - [h(F(x_k+s_k)) - h(m(x_k+s_k))]|}{\Psi^m(x_k,\Delta_k)} \;\leq\; \frac{(2L_h\kappa_{ef})\Delta_k^2}{\min\{\Delta_k,1\}\Psi_k^m}.$$

If $\Delta_k \leq 1$, then, from $\Delta_k \leq C_4\Psi_k^m$,

$$|\rho_k - 1| \;\leq\; \frac{(2L_h\kappa_{ef})\Delta_k}{\Psi_k^m} \;\leq\; \frac{(2L_h\kappa_{ef})C_4\Psi_k^m}{\Psi_k^m} \;=\; 1-\eta_1.$$

If $\Delta_k > 1$, then, from $\Delta_k \leq \sqrt{C_4\Psi_k^m}$

$$|\rho_k - 1| \;\leq\; \frac{(2L_h\kappa_{ef})\Delta_k^2}{\Psi_k^m} \;\leq\; \frac{(2L_h\kappa_{ef})C_4\Psi_k^m}{\Psi_k^m} \;=\; 1-\eta_1.$$

We then obtain $\rho_k \geq \eta_1$ implying $k \in \mathcal{S}$, which contradicts $k \in \mathcal{A}\cup\mathcal{U}$. Thus, (12) is not true. Now, from (11) and the fact that k is not in $\mathcal{C}$,

$$\Psi_k \;\leq\; |\Psi_k - \Psi_k^m| + \Psi_k^m \;\leq\; \kappa_\Psi\Delta_k + \Psi_k^m \;\leq\; (\kappa_\Psi\lambda + 1)\,\Psi_k^m,$$

and thus, $\Psi_k^m \geq \Psi_k/(\kappa_\Psi\lambda+1)$. Hence, since $\Delta_k \geq \min\{\sqrt{C_4\Psi_k^m}, C_4\Psi_k^m\}$, we have

$$\Delta_k \;\geq\; \frac{\min\{\sqrt{C_4\Psi_k}, C_4\Psi_k\}}{\kappa_\Psi\lambda+1}.$$

If the reduction in the trust-region radius occurs in the criticality step, then similarly to the last part of the proof of Lemma 3.2 (with $\|\nabla f(x_k)\|$, $\|g_k\|$, and κ_{eg} replaced by Ψ_k, Ψ_k^m, and κ_Ψ, respectively), it can be shown that $\Delta_k \geq \Psi_k/(\kappa_\Psi + 1/\beta)$. ■

A global convergence result can then be easily proved at this point of the analysis.

Theorem 5.1 *Let Assumptions 2.1 and 5.1 hold. Then* $\liminf_{k\to+\infty} \Psi_k = 0$.

Proof. By Lemma 5.2, there is an infinite subsequence of iterations where the trust-region radius is reduced, to which then we can apply Lemma 5.3. ■

5.3 Worst case complexity

We proceed by stating the analog of Lemma 3.3.

Lemma 5.4 *Let Assumptions 2.1 and 5.1 hold. Let* $\epsilon \in (0,1)$. *Let* k_0 *be the first iteration where* Δ_k *is reduced. For every iteration* $k \geq k_0$ *of the algorithm, if* $\Psi_j > \epsilon$ *for* $j = k_0, \ldots, k$, *then*

$$\Delta_k \;\geq\; \gamma C_5 \epsilon.$$

where $C_5 = \min\left\{\frac{\min\{\sqrt{C_4}, C_4\}}{\kappa_\Psi \lambda + 1}, \frac{1}{\kappa_\Psi + 1/\beta}\right\}$ *and* C_4 *is given in Lemma 5.3.*

Proof. When $k \in \mathcal{R}$, it follows directly from Lemma 5.3, $\Psi_k > \epsilon$, and $\epsilon < 1$, that $\Delta_k \geq C_5\epsilon$. When $k \notin \mathcal{R}$, the argument is the same as in the last paragraph of the proof of Lemma 3.3. ■

Again, to count the total number of iterations first we start by counting the number of successful iterations.

Theorem 5.2 *Let Assumptions 2.1 and 5.1 hold. Let* k_0 *be the index of the first iteration where* Δ_k *is reduced (which must exist from Lemma 5.2). Given any* $\epsilon \in (0,1)$, *assume that* $\Psi_{k_0} > \epsilon$ *and let* $\bar{k}$ *be the first iteration after* k_0 *such that* $\Psi_{\bar{k}} \leq \epsilon$. *Then, to achieve* $\Psi_{\bar{k}} \leq \epsilon$, *starting from* k_0, *Algorithm 5.1 takes at most* $|\mathcal{S}(k_0, \bar{k})|$ *successful iterations, where*

$$|\mathcal{S}(k_0, \bar{k})| \leq \frac{\lambda(f(x_{k_0}) - f_{low})}{\eta_1 \min\{\gamma C_5, 1\}\gamma C_5}\epsilon^{-2},$$

C_5 *is given in Lemma 5.4, and* $\mathcal{S}(k_0, \bar{k})$ *includes* k_0 *but excludes* $\bar{k}$.

Proof. Let $k \geq k_0$ be the index of a successful iteration. Using Lemma 5.1, $\Psi_k^m \geq \Delta_k/\lambda$, Lemma 5.4, and $\epsilon \in (0,1)$, we obtain

$$\begin{aligned} f(x_k) - f(x_{k+1}) \;&\geq\; \eta_1 \Psi^m(x_k, \Delta_k) \;\geq\; \eta_1 \min\{\Delta_k, 1\}\Psi_k^m \\ &\geq\; \eta_1 \min\{\Delta_k, 1\}\frac{\Delta_k}{\lambda} \;\geq\; \frac{\eta_1}{\lambda}\min\{\gamma C_5\epsilon, 1\}\gamma C_5\epsilon \\ &\geq\; \frac{\eta_1}{\lambda}\min\{\gamma C_5, 1\}\gamma C_5\epsilon^2. \end{aligned}$$

We then obtain by summing up all the successful iterations starting at k_0 that

$$f(x_{k_0}) - f(x_{\bar{k}}) \;\geq\; |\mathcal{S}(k_0, \bar{k})|\frac{\eta_1}{\lambda}\min\{\gamma C_5, 1\}\gamma C_5\epsilon^2,$$

and the proof is completed. ■

Now, we count the number of iterations after k_0 that are not successful.

Theorem 5.3 *Let Assumptions 2.1 and 5.1 hold. Let k_0 be the index of the first iteration where Δ_k is reduced (which must exist from Lemma 5.2). Given any $\epsilon \in (0,1)$, assume that $\Psi_{k_0} > \epsilon$ and let $\bar{k}$ be the first iteration after k_0 such that $\Psi_{\bar{k}} \leq \epsilon$. Then, to achieve $\Psi_{\bar{k}} \leq \epsilon$, starting from k_0, Algorithm 5.1 takes at most $|\mathcal{N}(k_0, \bar{k})|$ other (not successful) iterations, where*

$$|\mathcal{N}(k_0, \bar{k})| \ \leq \ (3 + 4L_3)|\mathcal{S}(k_0, \bar{k})| + 4\left(L_4 - \log_\gamma(e)\epsilon^{-1}\right),$$

where C_5 is given in Lemma 5.4, $L_3 = -\log_\gamma(\gamma_{inc})$, and $L_4 = \log_\gamma\left(\frac{\gamma^2 C_5 e}{\Delta_{k_0}}\right)$.

Proof. The proof, except using Lemma 5.4 instead of Lemma 3.3, follows along the lines of that of Theorem 3.2. ■

The number of iterations necessary to achieve the first iteration k_0 (where the trust-region radius is reduced) is $\mathcal{O}(1)$, and thus k_0 is of the order of $\mathcal{O}\left(\epsilon^{-2}\right)$, and the explanation is similar to the one for the smooth case discussed after Theorem 3.2. Again, as we saw in previous sections, some of the constants appearing in the bound on the number of iterations depend on the dimension of the problem space and on Lipschitz constants of first-order derivatives. In the case of this section we frame this dependence in the following assumption, which can be easily met if the model of F is formed by $F(x_k) + J^m(x_k)s$ where the transposed rows of $J^m(x_k)$ are computed as simplex gradients for the entries of F centered at x_k.

Assumption 5.2 *The constants κ_{ef} and κ_{eg} in the definition of fully linear models satisfy $\kappa_{ef} = \mathcal{O}(\sqrt{n}L_{J_F})$ and $\kappa_{eg} = \mathcal{O}(\sqrt{n}L_{J_F})$, where n is the problem dimension and L_{J_F} is the largest of the Lipschitz constants of F_i, $i = 1, \ldots, \ell$.*

Theorem 5.4 *Let Assumptions 2.1, 5.1, and 5.2 hold. To drive $\Psi(\cdot, 1)$ below $\epsilon \in (0,1)$, Algorithm 5.1 takes at most $\mathcal{O}\left(n\epsilon^{-2}\right)$ iterations.*

Proof. The proof is similar to that of Theorem 3.3. ■

The dependence of the bound on L_{J_F} was omitted but is $L_{J_F}^2$ as in Theorem 3.3 when $p = 2$.

Corollary 5.1 *Let Assumptions 2.1, 5.1, and 5.2 hold. To drive $\Psi(\cdot, 1)$ below $\epsilon \in (0,1)$, Algorithm 5.1 takes at most $\mathcal{O}\left(\ell n^2 \epsilon^{-2}\right)$ function evaluations.*

It can then be seen that, in terms of ϵ, the bound on the number of function evaluations derived in this paper is better by a factor of $|\log \epsilon|$ than the bound $\mathcal{O}(|\log \epsilon|\epsilon^{-2})$ derived in [23].

6 A numerical illustration

We have compared the numerical behavior of Algorithm 4.1 (smoothing trust-region approach) and a variant of Algorithm 5.1 (composite trust-region approach) on a test set suggested in [26] consisting of 53 problems of the form $\min_{x \in \mathbb{R}^n} f(x) = \|F(x)\|_1$. In this test set, F varies among 22 nonlinear vector functions of the CUTEr collection [22] with $2 \leq n \leq 12$ and different initial points.

In the smoothing approach (`Sdfo-tr`) we used the trust-region implementation described in [1] for each smooth outer iteration. Algorithm 4.1 was run using $\mu_0 = 10^4$, $r(\mu) = \min(10^{-5}, \mu^2)$,

and the update $\mu_{k+1} = \mu_k/100$. The algorithm was stopped when μ_k reaches 10^{-4}, which, given the initial value for μ_0, resulted in doing five outer iterations ($k = 0, 1, 2, 3, 4$). The final iterate and trust-region radius of the previous outer iteration were provided as the starting one for the next.

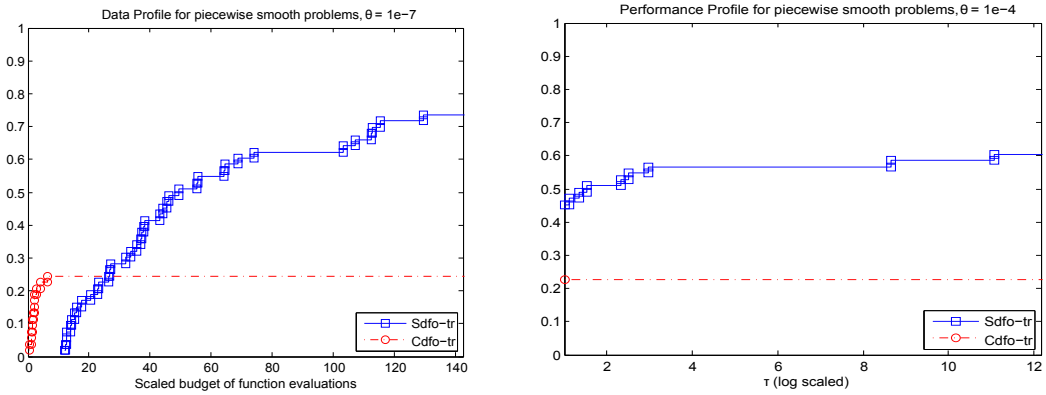
The same code from [1] was then adapted as the composite approach (`Cdfo-tr`), by changing the criticality measure and the trust-region subproblem. We used as models of F the linear ones $m_k(x_k + s) = F(x_k) + J^m(x_k)s$, where the transposed rows of $J^m(x_k)$ were regression simplex gradients computed using the $2n$ points $x_k \pm \mathrm{e}_i \min(10^{-2}, \Delta_k)$ (with e_i the ith coordinate vector). Since these models are always fully linear, no critical or model-improvement iterations were considered. The trust-region ball was defined using the ℓ_∞-norm so that the resulting trust-region subproblem was an LP (which was solved using the routine `linprog.m` from the Matlab Optimization Toolbox).

For both methods, we set the common initial parameters as $\Delta_{0,0} = 1$ (`Sdfo-tr`), $\Delta_0 = 1$ (`Cdfo-tr`), $\eta_0 = 10^{-3}$, $\eta_1 = 0.25$, $\gamma = 0.5$, $\gamma_{inc} = 1$ except when $\rho_k \geq 0.75$ where $\gamma_{inc} = 2$ and $\Delta_{max} = 10^3$. For `Sdfo-tr`, we set $p = 1.5$, $c_1 = 1$ and for `Cdfo-tr` we set $c_1 = 0$.

A data profile [26] is given in Figure 1(a), indicating the percentage of problems solved by the two methods under consideration as function of a budget of objective function evaluations (scaled by $n + 1$). A problem is considered solved when

$$f(x_0) - f(x) \;\geq\; (1 - \theta)[f(x_0) - f_L],$$

where $\theta \in (0, 1)$ is a level of accuracy, x_0 is the initial iterate, and f_L is the best objective value found by the two methods for a budget of 1500 function evaluations. The value of θ was set to 10^{-7}.



(a) Data profiles.

(b) Performance profiles in a logarithmic scale.

Figure 1: Performance and data profiles computed for a set of piecewise smooth problems, comparing the smoothing and composite trust-region methods.

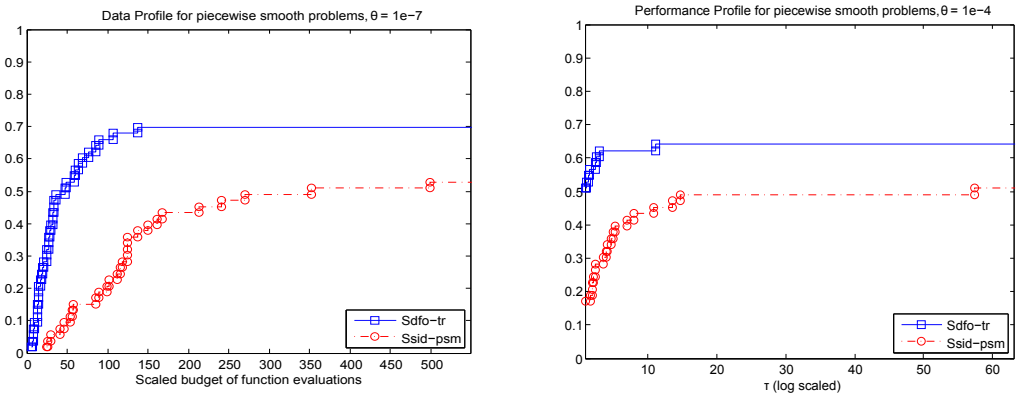
A performance profile [18] is then given in Figure 1(b), depicting how well a method performed relatively to the other in reaching the same (scale invariant) convergence test [19], in our case chosen as

$$f(x) - f_* \;\leq\; \theta(|f_*| + 1|),$$

where θ is the accuracy level and f_* is an approximation for the optimal value of the problem being tested. Each method curve describes (at $\tau = 1$) the fraction of problems for which the

method performs the best (efficiency) and (for τ sufficiently large) the fraction of problems solved by the method (robustness). The value of θ was set to 10^{-4} and the budget of function evaluations to 1500. The value of f_* was selected as the best value attained by these two methods and by those also tested in [16], to ensure that we indeed measure the real ability to solve the problems.

Despite the fact of exhibiting a worse WCC bound, the smoothing approach worked much better than the composite one, which does not come as a surprise given the absence of curvature exploration in the latter one. We then compared our smoothing trust-region approach with the smoothing direct search introduced in [21], on the same set of problems. Data and performance profiles are given in Figures 2(a) and 2(b), respectively, using the same levels of accuracy and budget of evaluations. It can be seen that the smoothing trust-region approach worked better, both in terms of efficiency and robustness.



(a) Data profiles.

(b) Performance profiles in a logarithmic scale.

Figure 2: Performance and data profiles computed for a set of piecewise smooth problems comparing the smoothing trust-region and direct-search methods.

7 Conclusions

This paper presented a unified coverage of the worst case complexity (WCC) of derivative-free trust-region methods for unconstrained optimization, from the case where the function is smooth to the case where it is non-smooth. In the non-smooth setting, we considered the general case of Lipschitz continuity and the case of a composite type structure. The WCC bounds established in the various cases were the expected ones, matching existent bounds for derivative-free or derivative-based optimization. The novelty of the paper consisted of the way under which the trust-region algorithms were analyzed, individually and all together.

The analysis of WCC of this paper can be refined along several ways. One possibility would be to establish a power of -1 in ϵ when f is convex and smooth. Another possibility is to measure the effort in approaching second-order stationary points (which has already been done in the Ph.D. thesis of the second author). Some extensions to constraints may be doable using the methodology of this paper.

Appendix

Proof of Lemma 3.1: *First we assume that the number of successful iterations is finite. Suppose that the number of iterations in $\mathcal{R}_\gamma = \mathcal{C}_\gamma^r \cup \mathcal{A} \cup \mathcal{U}$ is also finite. Then we would have an infinite number of iterations either in $\mathcal{C}_\gamma^{nr} = \mathcal{C} \setminus \mathcal{C}_\gamma^r$ or in $\mathcal{M}$. In the first case, a contradiction would be reached since after each iteration in $\mathcal{C}_\gamma^{nr}$ (a last in a series of critical ones) the model is fully linear and we would either have an iteration in $\mathcal{S}$, $\mathcal{A}$, or in $\mathcal{U}$. In the second case, since after a model-improvement iteration we have an iteration of different type, this would imply an infinite number of iterations in $\mathcal{C}_\gamma^r$, $\mathcal{C}_\gamma^{nr}$, $\mathcal{S}$, $\mathcal{A}$, or $\mathcal{U}$, which is not possible. Thus, there is an infinite number of iterations in $\mathcal{R}_\gamma = \mathcal{C}_\gamma^r \cup \mathcal{A} \cup \mathcal{U}$. Hence, Δ_k is decreased an infinite number of times by a factor of γ, which leads to the convergence of Δ_k to zero.*

Let us assume now that $\mathcal{S}$ is infinite. When k is in $\mathcal{S}$, using the bound on the fraction of Cauchy decrease (4), Assumption 3.2, and $\|g_k\| \geq \Delta_k/\lambda$ (since k is not critical),

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta_1 \kappa_{fcd}}{2\lambda} \Delta_k \min\left\{ \frac{\Delta_k}{\kappa_{bhm}\lambda}, \Delta_k \right\} + c_1 \Delta_k^p. \tag{13}$$

Given that $\mathcal{S}$ is considered infinite and f is assumed bounded from below, the right-hand side of (13) has to converge to zero for $k \in \mathcal{S}$. Hence $\lim_{k \in \mathcal{S}} \Delta_k = 0$, and the proof is completed when there are only successful iterations. The proof can be easily concluded as in the proof of Lemma 5.5 in [14].

Proof of Lemma 3.2: *By assumption we have that either k is in $\mathcal{R}_\gamma = \mathcal{C}_\gamma^r \cup \mathcal{A} \cup \mathcal{U}$ or $k \in \mathcal{C}_\gamma^{nr} = \mathcal{C} \setminus \mathcal{C}_\gamma^r$ and $\gamma \Delta_k < \beta \|g_{k+1}\| < \Delta_k$.*

Let us suppose that $k \in \mathcal{A} \cup \mathcal{U}$. We will show first that $\|g_k\| \leq C_0 \Delta_k + C_2 \Delta_k^{p-1}$. For that, suppose by contradiction that this inequality is false, i.e., that

$$\|g_k\| \;>\; C_0 \Delta_k + C_2 \Delta_k^{p-1}. \tag{14}$$

Using (4) and $C_0 \geq \kappa_{bhm}$, one has

$$m_k(x_k) - m_k(x_k + s_k) \;\geq\; \frac{\kappa_{fcd}}{2} \|g_k\| \min\left\{ \frac{\|g_k\|}{\kappa_{bhm}}, \Delta_k \right\} \;\geq\; \frac{\kappa_{fcd}}{2} \|g_k\| \Delta_k. \tag{15}$$

Hence, we have

$$\begin{aligned}
|\rho_k - 1| \;&=\; \left| \frac{f(x_k) - f(x_k + s_k) - c_1 \Delta_k^p}{m_k(x_k) - m_k(x_k + s_k)} - 1 \right| \\
&=\; \left| \frac{f(x_k) - f(x_k + s_k) - c_1 \Delta_k^p - m_k(x_k) + m_k(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \right| \\
&\leq\; \left| \frac{f(x_k + s_k) - m_k(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \right| + \left| \frac{f(x_k) - m_k(x_k)}{m_k(x_k) - m_k(x_k + s_k)} \right| \\
&\quad + \left| \frac{c_1 \Delta_k^p}{m_k(x_k) - m_k(x_k + s_k)} \right| \\
&\leq\; \frac{4 \kappa_{ef} \Delta_k}{\kappa_{fcd} \|g_k\|} + \frac{c_1 \Delta_k^{p-1}}{\frac{\kappa_{fcd}}{2} \|g_k\|} \;\leq\; 1 - \eta_1,
\end{aligned}$$

where the second inequality holds because of the fully linearity of the model (3) and of inequality (15), and the third inequality comes from (14) and $C_0 \geq 8\kappa_{ef}/\kappa_{fcd}(1 - \eta_1)$. Therefore, we

have $\rho_k \geq \eta_1$, implying that the iteration is successful and contradicting the fact that $k \in \mathcal{A} \cup \mathcal{U}$. We have thus proved that (14) is false. To establish the result of the lemma when $k \in \mathcal{A} \cup \mathcal{U}$, it remains to use (2) and write (adding and subtracting g_k)

$$\|\nabla f(x_k)\| \leq \kappa_{eg}\Delta_k + C_0\Delta_k + C_2\Delta_k^{p-1} = C_1\Delta_k + C_2\Delta_k^{p-1}.$$

It remains to consider the case where the reduction occurs in the criticality step. If k is not the last critical iteration in a series of them, then $\Delta_{k+1} = \gamma\Delta_k$ ($k \in \mathcal{C}_\gamma^r$) and $\Delta_{k+1} > \lambda\|g_{k+1}\|$. Thus (adding and subtracting now g_{k+1}),

$$\begin{aligned}\|\nabla f(x_k)\| &= \|\nabla f(x_{k+1})\| \leq \kappa_{eg}\Delta_{k+1} + \|g_{k+1}\| \\ &\leq \kappa_{eg}\gamma\Delta_k + \frac{\gamma\Delta_k}{\lambda} \leq \left(\kappa_{eg} + \frac{1}{\beta}\right)\gamma\Delta_k \leq C_1\Delta_k + C_2\Delta_k^{p-1}.\end{aligned}$$

If k is the last critical iteration in a series of them, then due to $\Delta_{k+1} = \min\{\Delta_k \max\{\gamma\Delta_k, \beta\|g_{k+1}\|\}\}$ and the assumed reduction in the trust-region radius, either $\Delta_{k+1} = \gamma\Delta_k \geq \beta\|g_{k+1}\|$ ($k \in \mathcal{C}_\gamma^r$) or $\gamma\Delta_k < \Delta_{k+1} = \beta\|g_{k+1}\| < \Delta_k$. In the first case we have $\|g_{k+1}\| \leq \gamma\Delta_k/\beta \leq \Delta_k/\beta$ and in the second case we have $\|g_{k+1}\| \leq \Delta_k/\beta$. Thus, again,

$$\begin{aligned}\|\nabla f(x_k)\| &= \|\nabla f(x_{k+1})\| \leq \kappa_{eg}\Delta_{k+1} + \frac{\Delta_k}{\beta} \\ &\leq \kappa_{eg}\Delta_k + \frac{\Delta_k}{\beta} = \left(\kappa_{eg} + \frac{1}{\beta}\right)\Delta_k \leq C_1\Delta_k + C_2\Delta_k^{p-1}.\end{aligned}$$

Proof of Lemma 3.3: *Let $k \geq k_0$ be an iteration where Δ_k is reduced. When $\Delta_k < 1$, by applying Lemma 3.2,*

$$\epsilon < (C_1 + C_2)\max\{\Delta_k, \Delta_k^{p-1}\} \leq (C_1 + C_2)\Delta_k^{\min(p-1,1)}.$$

If $\Delta_k \geq 1$, then $\Delta_k \geq \epsilon$. Hence, considering both cases of $\Delta_k < 1$ and $\Delta_k \geq 1$, and the fact that $\epsilon < 1$, we have $\Delta_k \geq C_3\epsilon^{1/\min(p-1,1)}$. The lemma is proved for all iterations $k \in \mathcal{R}$ such that $k \geq k_0$.

At iterations in $\mathcal{R} = \mathcal{C}^r \cup \mathcal{A} \cup \mathcal{U}$, Δ_k is decreased by a factor of at most γ. At iterations in $\mathcal{C}^{nr} \cup \mathcal{S} \cup \mathcal{M}$, Δ_k is not decreased. Thus, we can backtrack from any iteration k in $\mathcal{C}^{nr} \cup \mathcal{S} \cup \mathcal{M}$, to the previous iteration in $\mathcal{R}$, say k_1 (possibly $k_1 = k_0$), and obtain $\Delta_k \geq \gamma\Delta_{k_1}$.

Acknowledgement

We are grateful to two Referees for their careful reading and for a number of comments which improved the presentation of the paper.

References

[1] A. S. Bandeira, K. Scheinberg, and L. N. Vicente. Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization. *Math. Program.*, 134:223–257, 2012.

[2] J.V. Burke and T. Hoheisel. Epi-convergent smoothing with applications to convex composite functions. *SIAM J. Optim.*, 23:1457–1479, 2013.

[3] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization. *SIAM J. Optim.*, 20:2833–2852, 2010.

[4] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. *SIAM J. Optim.*, 21:1721–1739, 2011.

[5] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic overestimation methods for unconstrained optimization. Part II: worst-case function-evaluation complexity. *Math. Program.*, 130:295–319, 2012.

[6] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM J. Optim.*, 22:66–86, 2012.

[7] C. Chen and O. L. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Comput. Optim. Appl.*, 5:97–138, 1996.

[8] X. Chen. Smoothing methods for nonsmooth, nonconvex minimization. *Math. Program.*, 134:71–99, 2012.

[9] X. Chen and C. T. Kelley. Sampling methods for objective functions with embedded Monte Carlo simulations. 2013.

[10] X. Chen and W. Zhou. Smoothing nonlinear conjugate gradient method for image restoration using nonsmooth nonconvex minimization. *SIAM J. Imaging Sciences*, 3:765–790, 2010.

[11] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2000.

[12] A. R. Conn, K. Scheinberg, and Ph. L. Toint. On the convergence of derivative-free methods for unconstrained optimization. In M. D. Buhmann and A. Iserles, editors, *Approximation Theory and Optimization, Tributes to M. J. D. Powell*, pages 83–108. Cambridge University Press, Cambridge, 1997.

[13] A. R. Conn, K. Scheinberg, and L. N. Vicente. Geometry of interpolation sets in derivative free optimization. *Math. Program.*, 111:141–172, 2008.

[14] A. R. Conn, K. Scheinberg, and L. N. Vicente. Global converence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM J. Optim.*, 20:387–415, 2009.

[15] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[16] A. L. Custódio, H. Rocha, and L. N. Vicente. Incorporating minimum Frobenius norm models in direct search. *Comput. Optim. Appl.*, 46:265–278, 2010.

[17] M. Dodangeh and L. N. Vicente. Worst case complexity of direct search under convexity. *Math. Program.*, 155:307–332, 2016.

[18] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[19] E. D. Dolan, J. J. Moré, and T. S. Munson. Optimality measures for performance profiles. *SIAM J. Optim.*, 16:891–909, 2006.

[20] G. Fasano, J. L. Morales, and J. Nocedal. On the geometry phase in model-based algorithms for derivative-free optimization. *Optim. Methods Softw.*, 24:145–154, 2009.

[21] R. Garmanjani and L. N. Vicente. Smoothing and worst case complexity for direct-search methods in non-smooth optimization. *IMA J. Numer. Anal.*, 33:1008–1028, 2013.

[22] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Software*, 29:373–394, 2003.

[23] G. N. Grapiglia, J. Yuan, and Y. Yuan. A derivative-free trust-region algorithm for composite nonsmooth optimization. *Comp. Appl. Math.*, 2014, to appear.

[24] S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM J. Optim.*, 19:414–444, 2008.

[25] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.*, 45:385–482, 2003.

[26] J. J. Moré and S. M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM J. Optim.*, 20:172–191, 2009.

[27] Y. Nesterov. *Introductory Lectures on Convex Optimization.* Kluwer Academic Publishers, Dordrecht, 2004.

[28] Y. Nesterov. Random gradient-free minimization of convex functions. Technical Report 2011/1, CORE, 2011.

[29] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton's method and its global performance. *Math. Program.*, 108:177–205, 2006.

[30] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis.* Springer, Berlin, 1997, third printing in 2009.

[31] K. Scheinberg and Ph. L. Toint. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM J. Optim.*, 20:3512–3532, 2010.

[32] L. N. Vicente. Worst case complexity of direct search. *EURO Journal on Computational Optimization*, 1:143–153, 2013.

[33] Y. Yuan. Conditions for convergence of trust region algorithms for nonsmooth optimization. *Math. Program.*, 31:220–228, 1985.

[34] Y. Yuan. Recent advances in trust region algorithms. *Math. Program.*, 151:249–281, 2015.

[35] C. Zhang and X. Chen. Smoothing projected gradient method and its application to stochastic linear complementarity problems. *SIAM J. Optim.*, 20:627–649, 2009.