Diogo Júdice

Trust-Region Methods without using Derivatives: Worst-Case Complexity and the Non-Smooth Case

Tese de Doutoramento do Programa Inter-Universitário de Doutoramento em Matemática, orientada pelo Professor Doutor Luís Nunes Vicente e apresentada ao Departamento de Matemática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Julho de 2015

· U C ·

Universidade de Coimbra

Trust-Region Methods without using Derivatives: Worst Case Complexity and the Non-Smooth Case

Diogo Júdice

UC|UP Joint PhD Program in Mathematics

Programa Inter-Universitário de Doutoramento em Matemática

PhD Thesis | Tese de Doutoramento

December 2015

Acknowledgements

I would like to express my deepest and sincere gratitude to Professor Luís Nunes Vicente. Since the beginning of this journey, he has always pointed me in the right direction, both scientifically and personally. It would have been impossible to fulfill this task without him. I believe that I am now a better mathematician and a better person, mostly because of his guidance.

I would like to thank Dr. R. (Nima) Garmanjani for his help regarding the numerical experiments and his revision of this dissertation. I would also like to thank also Mrs Rute Andrade for having helped me in so many administrative matters.

I am grateful to Fundação para a Ciência e Tecnologia for having given me financial support (scholarship SFRH/BD/74401/2010) for my doctoral studies. I would also like to thank the University of Coimbra and, in particular, the Department of Mathematics, for all the academic, administrative, and technical support.

I would like to thank my parents and my siblings for everything. Without them, it would have been impossible to accomplish this task. In particular, I would like to thank my parents for all their help and support during this process. I also would like to thank my friends Miguel Côrte-Real and Benedita Garrett for their support and encouragement in difficult times.

Finally, I would like to dedicate this thesis to my daughters Maria and Matilde and to my brother Pedro.

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 thesis, we start by analyzing the worst case complexity of trust-region derivative-free methods for smooth functions (based on a modification of the existent general methodology), bounding the number of iterations and function evaluations to reach a certain threshold of first or second order stationarity.

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 using the general modified methodology of the smooth case.

Resumo

Os métodos de região de confiança formam uma classe geral de métodos para optimização contínua que encontram aplicação numa variedade de problemas e contextos. Em particular, estes métodos têm sido estudados e aplicados a problemas sem recurso a derivadas.

A análise dos métodos de região de confiança sem derivadas tem incidido em convergência global, mostrando que estes métodos geram sequências de pontos convergindo para pontos estacionários, independentemente do ponto inicial. Uma grande parte desta análise é feita no caso suave, sabendo-se pouco sobre a complexidade ou taxa global destes métodos. Nesta tese, começamos por analisar a complexidade no pior dos casos de métodos de região de confiança sem derivadas para funções suaves (recorrendo a uma modificação da metodologia geral existente), limitando o número de iterações e de avaliações de função necessárias para atingir uma determinada proximidade a estacionaridade de primeira ou segunda ordem.

Para o caso não suave, propomos uma abordagem de suavização, para a qual provamos convergência global e limitamos a complexidade no pior dos casos. Para o caso especial de funções não suaves resultantes da composição de funções suaves com funções não suaves e convexas, mostramos como melhorar os resultados existentes na literatura utilizando a metodologia geral modificada do caso suave.

Table of contents

List of figures

List of tables

Chapter 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 [18] and the recent survey paper by Yuan [69]).

This thesis 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 simulation. DFO has also been relatively well studied (see the book by Conn, Scheinberg, and Vicente [23]). 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 [20] 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 [19, 22]) 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 [32], where good numerical results were reported for an interpolation-based trust-region method which ignores the geometry of the sample sets. Scheinberg and Toint [63] 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 [23, 46]). 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. Such results are derived independently of the starting point which justifies saying that the rates are global.

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 [52] started by showing that the gradient or steepest descent method takes a number of iterations of the order of ε^{-2} — and we write that as $\mathcal{O}(\varepsilon^{-2})$ — to drive the norm of the gradient of the objective function below ε. This effort is reduced to $\mathcal{O}(\varepsilon^{-1})$ in the presence of convexity. It is known that such a bound is sharp or tight (see the example of Cartis, Gould, and Toint [11]). A similar worst case complexity bound of $\mathcal{O}(\varepsilon^{-2})$ has been proved by Gratton, Sartenaer, and Toint [41] for trust-region methods. The worst case complexity (WCC) bound on the number of iterations can be reduced to $\mathcal{O}(\varepsilon^{-1.5})$ for cubic overestimation methods (see Nesterov and Polyak [54] and Cartis, Gould, and Toint [10]).

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 worst-case complexity bound, of $\mathcal{O}(\varepsilon^{-2})$, was derived by Vicente [64] for smooth functions, and later refined to $\mathcal{O}(\varepsilon^{-1})$ when the function is convex by Dodangeh and Vicente [28]. Garmanjani and Vicente [34], using a smoothing approach, have shown a WCC bound of $\mathcal{O}(|\log\varepsilon|\varepsilon^{-3})$ in the non-smooth case. Similar WCC bounds were derived, in expectation, by Nesterov [53] for his random Gaussian smoothing approach. Cartis, Gould, and Toint [14] have derived a WCC bound of $\mathcal{O}(\varepsilon^{-1.5})$ for their derivative-free adaptive cubic overestimation algorithm, but using finite differences to approximate derivatives.

1.3 The contribution of this thesis

In this thesis we address the worst case complexity of trust-region methods for unconstrained DFO. Our contributions are fourfold.

First we consider the smooth case and, as expected, derive a WCC bound of $\mathcal{O}(\varepsilon^{-2})$ for the number of iterations and $\mathcal{O}(n^2\varepsilon^{-2})$ 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}. 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 [22].

A second contribution is again in the smooth case but related to the WCC of derivative-free trust-region methods when determining second-order critical points. It is known that such methods globally converge to points satisfying the second-order necessary conditions [22]. It is also known that derivative-based trust-region methods require a number of the $\mathcal{O}(\max\{\varepsilon_g^{-2}\varepsilon_H^{-1}, \varepsilon_H^{-3}\})$ iterations

to determine a point where the norm of the gradient of the objective function is below ε_g and the smallest eigenvalue of the Hessian of the function is above $-\varepsilon_H$ (see Cartis, Gould, and Toint [13]). In this thesis, we prove a bound of $\mathcal{O}(\varepsilon^{-3})$, with $\varepsilon = \varepsilon_g = \varepsilon_H$, when no derivatives are used, and refine it as $\mathcal{O}(n^5\varepsilon^{-3})$ under certain assumptions for the corresponding number of functions evaluations. Very recently, it was proposed in [39] a direct-search method (that may use eigenvectors of approximated Hessians as directions), achieving the same WCC bounds in iterations and function evaluations.

Thirdly, we address the general non-smooth case, and develop a smoothing trust-region approach in the same vein as for direct search [34]. 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 $\mathcal{O}(|\log\varepsilon|\varepsilon^{-3})$ (for function evaluations, $\mathcal{O}(n^2|\log\varepsilon|\varepsilon^{-3})$). The knowledge of the contribution [34] has provided some guidance on how to obtain this result, but a lot had still to be done, from building all necessary blocks from the smooth case to assembling all components in the new context of trust regions.

The fourth 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 [37] but under a restrictive setting (relatively to the general scenario in [22]) and with sub-optimal results. Their complexity result in terms of function evaluations is of the form $\mathcal{O}(|\log\varepsilon|\varepsilon^{-2})$, where ours will be just $\mathcal{O}(\varepsilon^{-2})$. We were able to remove the factor $|\log\varepsilon|$ precisely from the way we count iterations in the criticality step. Further, contrary to [37], we do not impose a reduction of the trust-region radius on model-improvement iterations. In terms of function evaluations, our bound looks like $\mathcal{O}(\ell n^2\varepsilon^{-2})$, where ℓ is the number of functions components in F.

The author of this thesis is a co-author of the paper [33], under review in the SIAM Journal on Optimization, where the first, third, and fourth contributions of this thesis are reported.

1.4 Organization of the thesis and some terminology

This thesis is organized as follows. We start by reviewing trust-region methods with and without derivatives in Chapter 2, focusing on their global convergence properties. Then we review global rates for nonlinear optimization, with and without derivatives, in Chapter 3. Our first two contributions are described in Chapter 4: Section 4.1 for the WCC of derivative-free trust-region methods for determining first-order critical points of smooth functions; Section 4.2 likewise but for second-order critical points. Chapter 5 addresses the non-smooth case. In Section 5.3 we introduce and analyze the smoothing approach. The non-smooth composite case is handled in Section 5.4. At the end of this chapter (Section 5.5), we provide a numerical illustration of the latter two approaches for the case $\|F\|_1$. The thesis is ended in Chapter 6, with some conclusions and prospects of future work.

In the thesis we will refer often to rates of convergence, most of the times in a global sense (where, as opposed to a local sense, no assumption is made on the proximity of the starting point to stationarity). Let $\{x_k\}_{k\geq 0}$ be a sequence in $\mathbb{R}^n$ converging to x_*. Consider the corresponding real sequence defined by $r_k = \|x_k - x_*\|$. We say that $\{x_k\}_{k\geq 0}$ has a linear rate of convergence if there exists $\theta \in (0,1)$ such that $r_{k+1}/r_k \leq \theta$ for all k sufficiently large. For example, the sequence $\{(1/2)^k\}_{k\geq 0}$ converges linearly. The rate of convergence can be slower or faster than linear. For the former case, the sequence $\{x_k\}_{k\geq 0}$ converges sublinearly if the ratio r_{k+1}/r_k converges to 1 (while retaining the property that r_k tends to

zero). Examples of real sequences exhibiting sublinear rates to $x_* = 0$ that appear often in first-order methods for continuous optimization are $\{1/k^2\}_{k\geq 0}$, $\{1/k\}_{k\geq 0}$, and $\{1/\sqrt{k}\}_{k\geq 0}$. For the latter case, the sequence $\{x_k\}_{k\geq 0}$ converges superlinearly if the ratio r_{k+1}/r_k converges to 0 (an example being $\{(1/2)^{k^2}\}_{k\geq 0}$ with $x_* = 0$). Finally, we say that the sequence $\{x_k\}_{k\geq 0}$ converges quadratically if $r_{k+1}/r_k^2 \leq M$, for some $M > 0$. An example is given by the sequence $\{10^{-2^k}\}_{k\geq 0}$ that converges quadratically to $x_* = 0$. The rates described so far are the q-rates where the "q" stands for quotient, see [56, Chapter 9]. There are also the so-called r-rates (r of root). A sequence converges with an r-rate to x_* if r_k is bounded by a real sequence converging with a q-rate to 0. For instance, the rate of convergence of the sequence $\{x_k\}_{k\geq 0}$ is r-linear if $r_k \leq y_k$ and $\{y_k\}_{k\geq 0}$ converges linearly to $0 \in \mathbb{R}$. An example is given by the sequence defined by $x_k = (1/2)^k$ for k even and $x_k = 0$ for k odd. In both cases, q and r, what we have described are consequences of the original, more complicated definitions [56, Chapter 9].

In the WCC bounds, the notation $\mathscr{O}(A)$ 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 appropriate.

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.

Chapter 2

Derivative-free trust-region methods for smooth functions

In this chapter we will review the basic concepts of trust-region algorithms for the unconstrained minimization of a smooth function $f : \mathbb{R}^n \to \mathbb{R}$, with or without derivatives.

Section 2.1 is devoted to the basics of trust-region methods in the presence of derivatives. We use a simple trust-region method to describe the concepts involved, in particular the quadratic models and the trust-region subproblem. The algorithm described converges to first and second order stationary points. We will also comment on the local rate convergence of a Newton's method globalized by such trust-region scheme. This material is now classic and a more comprehensive coverage is given in the book by Conn, Gould, and Toint [18], in [55, Chapter 4], and in the survey papers [48, 69]. The proofs of the results stated in this section can be found in [18].

In Section 2.2 we start to address trust-region methods but without the use of derivatives. The necessary tools for global convergence of derivative-free trust-region algorithms are presented. We will show how interpolation and regression techniques can be used to build models with the desired properties. Such models will replace the quadratic models using derivatives defined in Section 2.1. The material is mostly taken from [23].

In Section 2.3 it is presented the derivative-free trust-region framework of [22] (see also [23, Chapter 10]). This framework includes a number of provisions for the absence of derivatives including, for instance, criticality and model-improvement steps. The global convergence properties will be stated and discussed.

In Section 2.4 other derivative-free trust-region approaches are referred and commented.

2.1 Introduction to trust-region methods for smooth functions

A typical trust-region method approximates the objective function by a quadratic model in a neighborhood or ball of the current iterate point. Then it minimizes the model in that neighborhood (trust region). If the model solution is a good approximation for the function f, then the step is taken and the radius of the ball (trust-region radius) is possibly increased. If not, the trust-region radius is shrunk and the model is minimized again. This process is repeated until some form of approximate stationarity is reached.

To be more specific, let x_k be the current iterate. A trust region is typically a set of the form

$$B(x_k;\Delta_k) = \{x \in \mathbb{R}^n : \|x - x_k\| \leq \Delta_k\},$$

where Δ_k is the trust-region radius. Assuming that f is continuously differentiable, one can approximate f in $B(x_k;\Delta_k)$ by a quadratic of the form:

$$m_k(x_k + s) = f_k + g_k^\top s + \frac{1}{2} s^\top H_k s, \tag{2.1}$$

where $f_k = f(x_k)$, $g_k = \nabla f(x_k) \in \mathbb{R}^n$, and $H_k \in \mathbb{R}^{n\times n}$ is a symmetric matrix. Observe that from Taylor's Theorem, we know that the difference between $m_k(x_k+s)$ and $f(x_k+s)$ is $\mathscr{O}(\|s\|^2)$, which is small for small s. When the function is twice continuously differentiable, the matrix H_k is seen as an approximation to the Hessian matrix $\nabla^2 f(x_k)$. If $H_k = \nabla^2 f(x_k)$, the difference between $m_k(x_k+s)$ and $f(x_k+s)$ becomes $\mathscr{O}(\|s\|^3)$. To obtain the next iterate we must find s as an approximate solution of the trust-region subproblem

$$\min_{s\in\mathbb{R}^n} m_k(x_k+s) \;=\; f_k + g_k^\top s + \frac{1}{2} s^\top H_k s \quad s.t. \quad \|s\| \leq \Delta_k.$$

Observe that if H_k is positive definite and $\|H_k^{-1} g_k\| \leq \Delta_k$, the exact solution of this problem is $s_k^H = -H_k^{-1} g_k$.

The minimizer of the model $m_k(x_k+s)$ subject to $\|s\| \leq \Delta_k$ along the steepest descent direction $-g_k = -\nabla f(x_k)$ is called the Cauchy step s_k^C. For the trust-region method to globally converge to first-order stationarity, the approximate solution s_k must give a decrease on $m_k(x_k+\cdot)$ as good as the Cauchy step. It can be shown that [55, Lemma 4.3]

$$m_k(x_k) - m_k(x_k + s_k^C) \;\geq\; \frac{1}{2}\|g_k\| \min\left\{\frac{\|g_k\|}{\|H_k\|}, \Delta_k\right\},$$

where we assume that $\frac{\|g_k\|}{\|H_k\|} = +\infty$ when $H_k = 0$. In fact, to guarantee global convergence to first-order stationary points we only need the step s_k to be as good as the Cauchy step in the sense of

$$m_k(x_k) - m_k(x_k + s_k) \;\geq\; \kappa_{fcd}(m_k(x_k) - m_k(x_k + s_k^C)), \tag{2.2}$$

for some constant $\kappa_{fcd} \in (0,1)$. Thus, the step s_k will satisfy

$$m_k(x_k) - m_k(x_k + s_k) \;\geq\; \frac{\kappa_{fcd}}{2}\|g_k\| \min\left\{\frac{\|g_k\|}{\|H_k\|}, \Delta_k\right\}. \tag{2.3}$$

The next step of the trust-region iteration is to measure the quality of the trial step s_k. For this matter, we compare the decrease in the model m_k, given by $m_k(x_k) - m_k(x_k+s_k)$, with the actual decrease in the function f, given by $f(x_k) - f(x_k+s_k)$. We then define the ratio

$$\rho_k = \frac{f(x_k) - f(x_k+s_k)}{m_k(x_k) - m_k(x_k+s_k)}. \tag{2.4}$$

The numerator in ρ_k is called the *actual reduction* and the denominator the *predicted reduction*. The value of ρ_k is used to accept or not the trial point given by $x_{k+1} = x_k + s_k$. First, observe from (2.2) that $m_k(x_k) - m_k(x_k + s_k) \geq 0$. So, if $\rho_k < 0$ the function f increases at the trial point and so it must be rejected and the trust-region radius decreased (and for the moment we consider that the same is done when $\rho_k \in [0, \eta)$ for small $\eta \in (0,1)$). On the other hand, if ρ_k is bigger than $\eta \in (0,1)$, it means that the model represents the function well in that trust region. In such a case, one accepts the trial step and possibly increase the trust-region radius. A trust-region method is thus an iterative method that starts with an initial guess and an initial trust-region radius. To proceed let us define it now formally.

Algorithm 2.1.1 Trust-region method (for smooth functions; first-order)

Initialization: Choose an initial point x_0 and an initial trust-region radius $\Delta_0 \in (0, \Delta_{max}]$ for some $\Delta_{max} > 0$. Construct the initial model $m_0(x_0 + s)$ as in (2.1). The constants $\eta \in (0,1)$, $\gamma_{inc} > 1$, and $\gamma \in (0,1)$ are given. Set $k = 0$.

Step 1 (step calculation): Compute a step s_k that sufficiently reduces the model m_k, in the sense of (2.2).

Step 2 (acceptance of the trial point): Compute $f(x_k + s_k)$ and ρ_k.

If $\rho_k \geq \eta$, then $x_{k+1} = x_k + s_k$ and the model of the form (2.1) is constructed at the new iterate x_{k+1} 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 3 (trust-region radius update): Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \min\{\gamma_{inc}\Delta_k, \Delta_{max}\}] & \text{if } \rho_k \geq \eta, \\ \{\gamma\Delta_k\} & \text{if } \rho_k < \eta. \end{cases}$$

Increment k by one and go to Step 1.

Global convergence

Now we review the global convergence properties of Algorithm 2.1.1. Consider the initial level set

$$L(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}. \tag{2.5}$$

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 $L(x_0)$, on which the function must be bounded below.

Assumption 2.1.1 *Assume that f is bounded below on $L(x_0)$, that is, there exists a constant f_{low} such that, for all $x \in L(x_0)$, $f(x) \geq f_{low}$.*

The function is also assumed smooth in $L(x_0)$.

Assumption 2.1.2 *Assume that f is continuously differentiable with Lipschitz continuous gradient (with constant $L_{\nabla f}$) in an open domain containing the set $L(x_0)$.*

As in the convergence of most trust-region methods, one needs to assume that the model Hessians are uniformly bounded.

Assumption 2.1.3 *There exists a constant $\kappa_{bhm} > 0$ such that, for all x_k generated by the algorithm,*

$$\|H_k\| \leq \kappa_{bhm}.$$

Next we formally state that the difference between the model and the function is $\mathcal{O}(\|s\|^2)$.

Lemma 2.1.1 *Let Assumptions 2.1.2 and 2.1.3 hold. Then*

$$|m_k(x_k + s_k) - f(x_k + s_k)| \leq \left(\frac{\kappa_{bhm} + L_{\nabla f}}{2}\right) \|s_k\|^2.$$

The following lemma says that, as long as the trial point is not stationary, if the trust-region radius is small enough relatively to the size of the gradient, then a successful iteration occurs in a finite number of steps and f can be further reduced. The result is proved by showing that $|\rho_k - 1| \leq 1 - \eta$, using (2.3) and Lemma 2.1.1.

Lemma 2.1.2 *Let Assumptions 2.1.2 and 2.1.3 hold and $\nabla f(x_k) \neq 0$. Then there exists a constant $C_1 > 0$ such that if $\Delta_k \leq C_1 \|\nabla f(x_k)\|$ then $\rho_k \geq \eta$ and iteration k is successful.*

Algorithm 2.1.1 is globally convergent to first-order stationary points in the sense of generating a subsequence of iterates driving the gradient of the function to zero as stated in the next theorem. The proof uses Lemma 2.1.2 and the fact that the function is bounded from below (Assumption 2.1.1).

Theorem 2.1.1 *Let $\{x_k\}$ be a sequence generated by Algorithm 2.1.1. Let Assumptions 2.1.1–2.1.3 hold. Then*

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

Note that since $\eta > 0$ in Algorithm 2.1.1, $\rho_k \geq \eta$ is a sufficient decrease condition. When $\eta = 0$, $\rho_k > 0$ is equivalent to $f(x_k + s_k) < f(x_k)$ which amounts to impose a simple decrease condition on function values. The result of Theorem 2.1.1 is also valid when $\eta = 0$ provided that the trust-region radius is reduced when $0 \leq \rho_k < \eta$. In other words, the step can be taken when $\rho_k \in [0, \eta)$, but the radius is reduced.

As it is stated, Algorithm 2.1.1 also verifies the following stronger result:

Theorem 2.1.2 *Let $\{x_k\}$ be a sequence generated by Algorithm 2.1.1. Let Assumptions 2.1.1–2.1.3 hold. Then*

$$\lim_{k \to +\infty} \|\nabla f(x_k)\| = 0.$$

There is a counter example [68] showing that the lim result of Theorem 2.1.2 might not hold for simple decrease even with the provisions given after Theorem 2.1.1. There are ways of imposing Theorem 2.1.2 for simple decrease by changing the way of updating the trust-region radius which will later be discussed in the context where derivatives are not used.

In the presence of second-order derivatives, the quadratic model can be constructed using $H_k = \nabla^2 f(x_k)$. In order to make the algorithm globally convergent to second-order points, the step s_k has then to satisfy additional requirements.

For such a purpose, let $\lambda_{\min}(H_k)$ be the smallest eigenvalue of H_k, assumed negative for a moment. Let s_k^E be an eigenvector associated with $\lambda_{\min}(H_k)$:

$$H_k s_k^E \;=\; \lambda_{\min}(H_k) s_k^E.$$

Suppose, without loss of generality, that $\|s_k^E\| = \Delta_k$ and $(s_k^E)^\top \nabla f(x_k) < 0$. It can be shown that $\tau = 1$ is the optimal solution of

$$\min_{\tau \geq 0} \; m_k(\tau s_k^E) \; s.t. \; \|\tau s_k^E\| \;\leq\; \Delta_k,$$

and it satisfies

$$m_k(x_k) - m_k(x_k + s_k^E) \;\geq\; -\frac{1}{2}\lambda_{\min}(H_k)\Delta_k^2. \tag{2.6}$$

The step s_k is then required to satisfy a condition called the fraction of the eigenvalue decrease: If $\lambda_{\min}(H_k) \;\geq\; 0$, we suppose that

$$m_k(x_k) - m_k(x_k + s_k) \;\geq\; \kappa_{fod}(m_k(x_k) - m_k(x_k + s_k^C)).$$

where $\kappa_{fod} \in (0,1)$. Otherwise, we suppose that

$$m_k(x_k) - m_k(x_k + s_k) \;\geq\; \kappa_{fod} \max\{m_k(x_k) - m_k(x_k + s_k^C), m_k(x_k) - m_k(x_k + s_k^E)\}, \tag{2.7}$$

The structure of the second-order algorithm is similar to Algorithm 2.1.1. A main difference is that now H_k is set to the Hessian of f at x_k. Another difference is in the trust-region subproblem solution: it has to satisfy (2.7).

Algorithm 2.1.2 Trust-region method (for smooth functions; second-order)

Initialization: Choose an initial point x_0 and an initial trust-region radius $\Delta_0 \in (0, \Delta_{max}]$ for some $\Delta_{max} > 0$. Construct the initial model $m_0(x_0 + s)$ as in (2.1) with $H_0 = \nabla^2 f(x_0)$. The constants $\eta \in (0,1)$, $\gamma_{inc} > 1$, and $\gamma \in (0,1)$ are given. Set $k = 0$.

Step 1 (step calculation): Compute a step s_k that sufficiently reduces the model m_k, in the sense of (2.7).

Step 2 (acceptance of the trial point): Compute $f(x_k + s_k)$ and ρ_k.

If $\rho_k \geq \eta$, then $x_{k+1} = x_k + s_k$ and the model of the form (2.1) is constructed at the new iterate x_{k+1}, with $H_{k+1} = \nabla^2 f(x_{k+1})$ 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 3 (trust-region radius update): Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \min\{\gamma_{inc}\Delta_k, \Delta_{max}\}] & \text{if } \rho_k \geq \eta, \\ \{\gamma\Delta_k\} & \text{if } \rho_k < \eta. \end{cases}$$

Increment k by one and go to Step 1.

Now we review the global convergence of Algorithm 2.1.2 to second-order stationary points. First, we need to assume that the Hessian is Lipschitz continuous.

Assumption 2.1.4 *Assume that f is twice continuously differentiable with Lipschitz continuous Hessian (with constant $L_{\nabla^2 f}$ in an open set containing the set $L(x_0)$.*

We also need to assume that the Hessian is uniformly bounded.

Assumption 2.1.5 *There exists a constant κ_{bhm} such that, for all x_k generated by the algorithm,*

$$\|\nabla^2 f(x_k)\| \leq \kappa_{bhm}.$$

Next we formally state that the difference between the model and the function is $\mathcal{O}(\|s\|^3)$.

Lemma 2.1.3 *Let Assumptions 2.1.4 and 2.1.5 hold. Then*

$$|m_k(x_k+s_k) - f(x_k+s_k)| \leq \left(\frac{L_{\nabla^2 f}}{6}\right)\|s_k\|^3.$$

The following lemma is a second-order version of Lemma 2.1.2. It says that, as long as the point is not second-order stationary, if the trust-region radius is small enough relatively to the size of the second-order stationarity measure $\sigma(x_k)$, where

$$\sigma(x) = \max\{\|\nabla f(x)\|, -\lambda_{\min}(\nabla^2 f(x))\}, \tag{2.8}$$

then a successful iteration occurs in a finite number of steps and f can be further reduced.

Lemma 2.1.4 *Let Assumptions 2.1.4 and 2.1.5 hold and $\sigma(x_k) \neq 0$. Then there exists a constant $C_2 > 0$ such that if $\Delta_k \leq C_2\sigma(x_k)$ then $\rho_k \geq \eta$ and iteration k is successful.*

The next theorem establishes global convergence to second-order stationary points.

Theorem 2.1.3 *Let Assumptions 2.1.1, 2.1.4, and 2.1.5 hold. Let $\{x_k\}$ be a sequence generated by the algorithm, where $H_k = \nabla^2 f(x_k)$ and s_k satisfies a fraction of the eigenvalue decrease. Then*

$$\liminf_{k\to+\infty} \sigma(x_k) = 0.$$

It is well known that it is not possible to prove a lim-type result of the type

$$\lim_{k\to+\infty} \sigma(x_k) = 0 \tag{2.9}$$

for an algorithm of the type of Algorithm 2.1.2 without modifying the scheme that updates the trust-region radius in successful iterations. A known modification is to increase the trust-region radius in all successful iterations. In such a case, it is possible to prove a lim-type result of the form (2.9), see, for instance, [18, Theorem 6.6.7].

A trust-region scheme is a globalization procedure that enables Newton or quasi-Newton schemes to converge from arbitrary starting points. In fact, it is well known that such methods enjoy a fast local rate of convergence (quadratic in the case of Newton and superlinear for quasi-Newton methods), but such properties require the starting point to be near a point satisfying the second-order sufficient optimality conditions. Away from those points, steps from these methods can be too large and need to be restricted. Line searches and trust regions are the two main schemes for doing that.

However, a globalization scheme must be able to recognize the proximity of such a point and then become inactive. In trust-region methods that is encompassed by not reducing the radius after a certain order. Such a global/local behavior can be described by the following result.

Theorem 2.1.4 *Let f be twice continuously differentiable at x_* and $\nabla^2 f$ Lipschitz continuous near x_*. Let $\{x_k\}$ be a sequence generated by Algorithm 2.1.1, where (for sufficiently large k) $H_k = \nabla^2 f(x_k)$ and $s_k = s_k^N$ when $\|s_k^N\| \leq \Delta_k$, and $s_k^N = -\nabla^2 f(x_k)^{-1}\nabla f(x_k)$ is well defined. Suppose that $\{x_k\}$ converges to point x_* and this one is such that $\nabla f(x_*) = 0$ and $\nabla^2 f(x_*)$ is positive definite.*

Then there exist $\Delta_* > 0$ and $k_* \in \mathbb{N}$ such that $\Delta_k \geq \Delta_*$ for all $k \geq k_*$.

Note that the assumptions of this theorem do not conflict with s_k satisfying a fraction of the Cauchy decrease. When $\|s_k^N\| \leq \Delta_k$ and $\nabla^2 f(x_k)$ is positive definite, s_k^N is the minimizer of the quadratic $m_k(x_k + s)$ subject to $\|s\| \leq \Delta_k$, and thus the decrease of s_k^N is larger in $m_k(x_k + s)$ than the decrease of the Cauchy step s_k^C.

As a consequence of the result of Theorem 2.1.4, the trust-region step becomes eventually the Newton one. In fact, since $\nabla f(x_k)$ converges to zero and $\nabla^2 f(x_k)$ converges to a positive definite matrix, s_k^N converges to zero. Then, the result of the Theorem says that the Newton step is inside the trust region for sufficiently large k and the modification of the algorithm stated in the theorem enables to take it. The local rate of convergence becomes then quadratic.

A similar result can be obtained for quasi-Newton type methods by taking a step s_k satisfying $\|s_k - s_k^N\| = o(s_k^N)$ when $\|s_k^N\| \leq \Delta_k$, yielding a superlinear rate of convergence.

2.2 Introduction to derivative-free trust-region concepts

When applying trust-region methods to problems where one can use derivatives of the objective function, we have access to the gradient and possibly to the Hessian. These objects are then used to build the quadratic models to be minimized in trust-region subproblems. In derivative-free trust-region methods, one only has access to function values, and the quadratic models must therefore strictly depend on the evaluation of the objective function on sample sets. Such models are typically built using interpolation or regression techniques and polynomial basis functions. The models must, however, enjoy the same accuracy properties as the Taylor based models used in the presence of derivatives (expressed in Lemmas 2.1.1 and 2.1.3). As we will see in this section, such an accuracy depends strongly on the geometrical properties of the sample sets.

Classical multivariate polynomial interpolation provides a measure for the quality of the geometry of the sample sets based on the corresponding notion of Lagrange polynomials. Such polynomials are defined in the space of polynomials used for the modeling in question. Each Lagrange polynomial is associated with a sample point, and thus they are as many as the number of points in a sample set.

Given a scenario where the interpolation is determined, i.e., where there are as many points as basis functions, each Lagrange polynomial is defined by the property that its value is equal to one at the corresponding point and to zero at the remaining ones. The maximal absolute value of all Lagrange polynomials in a compact set containing the sample set (or a bound Λ for that value) is a measure of its geometry, called the Lebesgue constant. The sample set in this case is called Λ-poised. Classical multivariate polynomial interpolation provides Taylor-type accuracy bounds between the function and the interpolating polynomial that depend on the Lebesgue constant. Lagrange polynomials can also be defined in the underdetermined and regression cases, where the cardinality of the sample set is less than or more than (respectively) the number of basis elements (see, respectively, [23, Chapter 5] and [23, Chapter 4]).

There is, however, an alternative and equivalent way of measuring the quality of sample sets for polynomial interpolation and regression, that is perhaps more intuitive and easier to monitor in certain numerical contexts and for which it is also possible to derive Taylor-type accuracy bounds for the corresponding polynomial models. This measure is essentially the condition number of the matrix appearing in the interpolation conditions, but for a sample set obtained from the original by first shifting and then scaling its points so that all the resulting points lie in the unitary ball centered at the origin. In the book [23] (originally in the papers [20, 21]), it is proved for all types of polynomial modeling (underdetermined, determined, and regression) that imposing a bound on such a condition number is equivalent to impose a bound on the maximum absolute value of the Lagrange polynomials (i.e., on being Λ-poised). In this section we will review polynomial modeling and the corresponding accuracy bounds using the condition number approach of [23].

Fully linear models

Let $x_0 \in \mathbb{R}^n$ be a starting point for the trust-region methods considered in this thesis. Let $f : \mathbb{R}^n \to \mathbb{R}$ 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 $L(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$, (see 2.5).

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:

$$L_{enl}(x_0) \;=\; \bigcup_{x \in L(x_0)} B(x;\Delta_{max}), \tag{2.10}$$

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 2.2.1 *Suppose x_0 and Δ_{max} are given. Assume that f is continuously differentiable with Lipschitz continuous gradient (with constant $L_{\nabla f}$) 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 [23].

Definition 2.2.1 *Let a function $f : \mathbb{R}^n \to \mathbb{R}$, that satisfies Assumption 2.2.1, be given. A set of model functions $M = \{m : \mathbb{R}^n \to \mathbb{R}, m \in C^1\}$ 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 gradient, and such that*

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

   $$\|\nabla f(x+s) - \nabla m(x+s)\| \leq \kappa_{eg}\Delta, \quad \forall s \in B(0;\Delta), \tag{2.11}$$

   and

   - *the error between the model and function satisfies*

   $$|f(x+s) - m(x+s)| \leq \kappa_{ef}\Delta^2, \quad \forall s \in B(0;\Delta). \tag{2.12}$$

   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 fully linear models are not necessarily linear, in fact they are typically quadratic in practice. Either way, the most popular models are based on polynomial basis functions. For this purpose, we start by reviewing basic concepts and notation for multivariate polynomial interpolation and regression.

General considerations

The model-improvement algorithms can be either based on the maximization of the absolute value of the Lagrange polynomials or on the use of pivotal algorithms over the interpolation matrices (see respectively Sections 6.2 and 6.3 of [23]). In this section we will only review how polynomial models can achieve the (Taylor-type) error bounds of the form (2.11) and (2.12).

Let us consider P_n^d, the space of polynomials in $\mathbb{R}^n$ of degree less or equal to d. The dimension of this space is $q_1 = n+1$ for $d = 1$ and $q_1 = (n+1)(n+2)/2$ for $d = 2$. Consider a basis for this space $\phi = \{\phi_0(x), \phi_1(x), ..., \phi_q(x)\}$, where $q_1 = q+1$. Elements of P_n^d can be written as $m(x) = \alpha^\top \phi(x)$,

with $\alpha \in \mathbb{R}^{q+1}$. Consider a set of sample points $Y = \{y^0, y^1, ..., y^p\} \subset \mathbb{R}^n$. We say that a polynomial $m \in P_n^d$ interpolates the function $f : \mathbb{R}^n \to \mathbb{R}$ at the set Y if

$$(m(y^i) =) \sum_{j=0}^{q} \alpha_j \phi_j(y^i) \;=\; f(y^i), \;\; i = 0, ..., p. \tag{2.13}$$

Conditions (2.13) form a linear system in terms of the interpolation coefficients α. In matrix form, it is equivalent to

$$M(\phi, Y)\alpha \;=\; f(Y), \tag{2.14}$$

where $f(Y) = \big(f(y^0), f(y^1), \ldots, f(y^p)\big)^\top$ and $M(\phi, Y)_{ij} = \phi_j(y^i)$, $i = 1, \ldots, p+1$, $j = 1, \ldots, q+1$. For the moment let us suppose that the interpolation matrix is square, which happens when there are as many sample points as basis functions (i.e., $p+1 = q+1$). For the interpolation system (2.14) to have a unique solution, the matrix $M(\phi, Y)$ has to be nonsingular. In this case, we say that the set Y is poised for polynomial interpolation in $\mathbb{R}^n$. Under these conditions the interpolating polynomial $m(x)$ exists and is unique (and is independent of the basis ϕ), see [23, Lemma 3.2].

Let us consider now the case when there are more points than basis functions, i.e., $p+1 > q+1$. In this case, one can compute a least-squares solution of (2.14), i.e., a minimizer of $\|M(\phi, Y)\alpha - f(Y)\|$, where, recall, $\|\cdot\|$ stands for the Euclidean norm. For that linear system (2.14) to have a unique solution in the least-squares sense, the matrix $M(\phi, Y)$ must have full column rank. In this case, we say that the set Y is poised for polynomial regression in $\mathbb{R}^n$. Under these conditions the regression polynomial $m(x)$ exists and is unique (and is independent of the basis ϕ), see [23, Lemma 4.3].

As mentioned earlier, a measure of the quality of the geometry of the sample set is given by the conditioning of the interpolation matrix for a shifted and scaled set. Given a set $Y = \{y^0, y^1, \ldots, y^p\}$, we first shift all the points by $-y^0$ so that the first new point will be the origin and then scale the remaining ones so that they lie in the unitary ball. In other words, we do

$$\hat{Y} \;=\; \{0, (y^1 - y^0)/\Delta, \ldots, (y^p - y^0)/\Delta\} \;\subseteq\; B(0; 1), \tag{2.15}$$

where $\Delta = \Delta(Y) = \max_{1 \le i \le p} \|y^i - y^0\|$. As we will see later for different scenarios, the Taylor-type accuracy bounds for the different interpolating or regression polynomials will depend on the conditioning of the matrix

$$\hat{M} \;=\; M(\bar{\phi}, \hat{Y}), \tag{2.16}$$

where $\bar{\phi}$ is the natural basis of P_n^d, or of particular submatrices or related submatrices.

Linear interpolation and regression models

We start by reviewing the linear case where $d = 1$ and $\bar{\phi} = \{1, x_1, \ldots, x_n\}$. In this scenario, one has

$$\hat{M} \;=\; \begin{bmatrix} 1 & 0 \\ e & \hat{L} \end{bmatrix},$$

where $e = (1,\ldots,1)^\top \in \mathbb{R}^n$ and

$$\hat{L} = \frac{1}{\Delta}[y^1 - y^0 \cdots y^p - y^0]. \tag{2.17}$$

First, let us consider the determined case $p = q = n$. In this case, the error bounds are derived in terms of the submatrix $\hat{L}$ under the following assumption.

Assumption 2.2.2 *Assume that the function f is continuously differentiable in an open domain Ω containing $B(y^0;\Delta(Y))$ and ∇f is Lipschitz continuous with constant $L_{\nabla f}$ in Ω.*

The following result is taken from [23, Theorems 2.11 and 2.12]. Note that in this case $m(x) = \alpha^\top \phi(x) = \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_n x_n$.

Theorem 2.2.1 *Let Assumption 2.2.2 hold and $d = 1$. Assume that the set Y is a poised set of sample points (in the determined sense) contained in the ball $B(y^0;\Delta(Y))$ of radius $\Delta = \Delta(Y)$. For all points $y \in B(y^0;\Delta)$, the following bounds hold*

$$\|\nabla f(y) - \nabla m(y)\| \leq L_{\nabla f}(1 + n^{\frac{1}{2}}\|\hat{L}^{-1}\|/2)\Delta,$$

$$|f(y) - m(y)| \leq L_{\nabla f}(3/2 + n^{\frac{1}{2}}\|\hat{L}^{-1}\|/2)\Delta^2.$$

A similar result exists for the overdetermined case, where the polynomial is obtained by least-squares regression. This is [23, Theorem 2.13], and we reproduce it below:

Theorem 2.2.2 *Let Assumption 2.2.2 hold and $d = 1$. Assume that the set Y is a poised set of sample points (in the regression sense) contained in the ball $B(y^0;\Delta(Y))$ of radius $\Delta = \Delta(Y)$. For all points $y \in B(y^0;\Delta)$, the following holds*

$$\|\nabla f(y) - \nabla m(y)\| \leq L_{\nabla f}(1 + p^{\frac{1}{2}}\|\hat{M}^\dagger\|/2)\Delta,$$

$$|f(y) - m(y)| \leq L_{\nabla f}(5/2 + p^{\frac{1}{2}}\|\hat{M}^\dagger\|)\Delta^2,$$

where $\hat{M}^\dagger = (\hat{M}^\top\hat{M})^{-1}\hat{M}^\top$ denotes the left inverse of $\hat{M}$.

The last two results show that we can build models using linear interpolation or linear regression on a poised sample set of points, satisfying the requirements (2.11) and (2.12) of fully linear models.

Several DFO methods are based on the notion of a simplex gradient. Given $n+1$ points, a simplex gradient is the gradient of the linear interpolation model $m(y) = \alpha^\top \phi(y) = c + g^\top y$, i.e., it is the vector g. Simplex gradients can be computed in a regression way when there are more than $n+1$ points, and are generally referred to as the gradient g of the linear regression model $m(y) = f(y^0) + (y - y^0)^\top g$ interpolating the first point y^0.

Underdetermined quadratic interpolation models

Linear models cannot incorporate the curvature of the function and are of limited use in trust-region methods. The most popular models are quadratic; using the previous notation, one has $d = 2$ and P_n^2. The natural basis in this space is $\bar{\phi} = \{1, x_1, \ldots, x_n, x_1^2/2, x_1x_2, \ldots, x_{n-1}x_n, x_n^2/2\}$. In the DFO context, many times the function that we want to minimize is costly to evaluate. As we saw earlier, we need $q_1 = (n+1)(n+2)/2$ points to build a complete interpolating model. If n is large, this can be prohibitive. This section explains how to build interpolating models when one has less than $(n+1)(n+2)/2$ points available. In such a case, the interpolating matrix $M(\phi, Y)$ has more columns than rows and the interpolating system (2.14) has no longer a unique solution. Hence, the corresponding interpolation polynomials are no longer unique.

What we are reviewing in this section are the ways of building models between these extreme cases, this is, using more than $n+1$ points (linear models) but less than $(n+1)(n+2)/2$ points (quadratic models) to try to use some curvature information to speed up convergence. Similarly to what we did before for linear interpolation and regression, one reviews here the results showing that we can build models using undetermined quadratic interpolation satisfying (2.11) and (2.12) of fully linear models (and ignore model-improvement algorithms).

We recall here the definition (2.17) of $\hat{L}$, and denote its left inverse by $\hat{L}^\dagger = (\hat{L}^\top \hat{L})^{-1}\hat{L}^\top$. The next theorem (see [23, Theorem 5.4]) says that we can build models satisfying the requirements (2.11) and (2.12) of fully linear models through quadratic undetermined interpolation, provided the norm of the Hessian of the models is bounded in some way. For convenience, we write the quadratic model $m(x) = \alpha^\top \phi(x) = f + g^\top x + \frac{1}{2}x^\top Hx$. One has $\nabla m(x) = g + Hx$ and $\nabla^2 m(x) = H$.

Theorem 2.2.3 *Let Assumption 2.2.2 hold. Assume that the set Y is a poised set of sample points (in the linear interpolation or regression sense if $p > n$) contained in the ball $B(y^0; \Delta(Y))$ of radius $\Delta = \Delta(Y)$. Then, for all points y in $B(y^0; \Delta(Y))$, we have that*

$$\|\nabla f(y) - \nabla m(y)\| \;\leq\; \tfrac{5\sqrt{p}}{2}\|\hat{L}^\dagger\|(L_{\nabla f} + \|H\|)\Delta,$$

$$|f(y) - m(y)| \;\leq\; \tfrac{5\sqrt{p}}{2}\|\hat{L}^\dagger\|(L_{\nabla f} + \|H\|)\Delta^2 + \tfrac{1}{2}(L_{\nabla f} + \|H\|)\Delta^2,$$

where H is the Hessian of the model.

This theorem provides schemes to compute models. In fact, one needs the models to have an Hessian with the minimum possible norm, and this leads to a number of possibilities that we will review next.

Minimum Frobenius norm models (for undetermined quadratic interpolation)

Let us consider again the natural basis $\bar{\phi}$ for P_n^2 and split it into linear and quadratic parts: $\bar{\phi}_L = \{1, x_1, \ldots, x_n\}$ and $\bar{\phi}_Q = \{\frac{1}{2}x_1^2, x_1x_2, \ldots, \frac{1}{2}x_n^2\}$. The interpolation model takes the form

$$m(x) \;=\; \alpha_L^\top \bar{\phi}_L(x) + \alpha_Q^\top \bar{\phi}_Q(x),$$

where α_L and α_Q are the appropriate parts of the coefficient vector α. We define the minimum Frobenius norm solution α^{mfn} as a solution of the following optimization problem in α_L and α_Q.

$$\begin{aligned} \min \ & \frac{1}{2}\|\alpha_Q\|^2 \\ s.t. \ & M(\bar{\phi}_L, Y)\alpha_L + M(\bar{\phi}_Q, Y)\alpha_Q \ = \ f(Y), \end{aligned} \tag{2.18}$$

where the matrix $M(\bar{\phi}, Y)$ has been considered in the blocks $M(\bar{\phi}_L, Y)$ and $M(\bar{\phi}_Q, Y)$. This is approximately equivalent to minimize the Frobenius norm of the Hessian H of $m(x)$. In fact, the Frobenius norm of H and the Euclidean norm of α lead to almost the same polynomials in the components of α_Q. Consider the simple example where $n = 2$ and $\alpha_Q = (\alpha_3, \alpha_4, \alpha_5)^\top$. The Hessian H is given by

$$H \ = \ \begin{bmatrix} \alpha_3 & \alpha_4 \\ \alpha_4 & \alpha_5 \end{bmatrix}.$$

As we know, $\|\alpha_Q\|^2 = \alpha_3^2 + \alpha_4^2 + \alpha_5^2$ and $\|H\|_F^2 = \alpha_3^2 + 2\alpha_4^2 + \alpha_5^2$. So, $\|\alpha_Q\| \neq \|H\|_F$ but the effect of minimizing $\|\alpha_Q\|$ is roughly the same as minimizing $\|H\|_F$.

The solution of the convex quadratic program (2.18) is given by its necessary optimality conditions, which in turn are equivalent to solving a linear system where the matrix is

$$F(\bar{\phi}, Y) \ = \ \begin{bmatrix} M(\bar{\phi}_Q, Y)M(\bar{\phi}_Q, Y)^\top & M(\bar{\phi}_L, Y) \\ M(\bar{\phi}_L, Y)^\top & 0 \end{bmatrix}. \tag{2.19}$$

If this matrix is nonsingular, then the minimum Frobenius norm model exists and it is unique. In this case, we say that the sample set Y is poised in the minimum Frobenius norm sense. This also implies poisedness in the linear interpolation or regression senses. Note that $F(\bar{\phi}, Y)$ is nonsingular if and only if $M(\bar{\phi}_L, Y)$ has full column rank and $M(\bar{\phi}_Q, Y)M(\bar{\phi}_Q, Y)^\top$ is positive definite in the null space of $M(\bar{\phi}_L, Y)^\top$.

The next result ([23, Theorem 5.7]) shows that the Hessian of the minimum Frobenius norm model is bounded, and, hence, these models satisfy the requirements (2.11) and (2.12) of fully linear models. Recall that a sample set is Λ-poised in a domain if the maximum absolute value of all Lagrange polynomials (in this case in the minimum Frobenius norm sense) in that domain are bounded by Λ.

Theorem 2.2.4 *Let Assumption 2.2.2 hold. Assume that the set Y is a Λ-poised set of sample points (in the minimum Frobenium norm sense) contained in the ball $B(y^0; \Delta(Y))$ of radius $\Delta = \Delta(Y)$. Given an upper bound Δ_{max} on Δ, we have that the Hessian H of the minimum Frobenius norm model satisfies*

$$\|H\| \ \leq \ \frac{4(p+1)\sqrt{q+1}L_{\nabla f}\Lambda}{c(\Delta_{\max})},$$

where $c(\Delta_{\max}) = \min\{1, 1/\Delta_{\max}, 1/\Delta_{\max}^2\}$.

As we mentioned earlier, we need models that have a reduced Hessian norm in order to promote models that satisfy the requirements (2.11) and (2.12) of fully linear models. Powell ([57, 58]) suggested to solve the undetermined interpolation system (2.14) by choosing the solution that minimizes the norm between the Hessian model H and a previous calculated Hessian model H^{old}. Such a model

is the solution of

$$\begin{array}{l} \min \dfrac{1}{2}\|H - H^{old}\|_F^2 \\ s.t.\ M(\bar{\phi}_L, Y)\alpha_L + M(\bar{\phi}_Q, Y)\alpha_Q \ = \ f(Y). \end{array} . \tag{2.20}$$

In a certain way, this resembles the spirit of quasi-Newton updates. Powell provided practical schemes to update such models ensuring good geometry and thus also error bounds like in the definitions of fully linear models.

Fully quadratic models

To establish global convergence to second-order stationary points (and the corresponding rates or complexity bounds) of the derivative-free trust region methods, certain models of f need to be assumed as accurate as second-order Taylor models, in the sense of Point 1 of the definition below. For that purpose, we need to assume that f is twice continuously differentiable.

Assumption 2.2.3 *Suppose x_0 and Δ_{max} are given. Assume that f is twice continuously differentiable with Lipschitz continuous Hessian (with constant $L_{\nabla^2 f}$) in an open domain containing the set $L_{enl}(x_0)$.*

It is further assumed that models in question can be made second-order accurate or *fully quadratic* in a finite number of model-improvement steps. We reproduce below Definition 10.4 in [23].

Definition 2.2.2 *Let a function f, that satisfies Assumption 2.2.3, be given. A set of model functions $M = \{m : \mathbb{R}^n \to \mathbb{R}, m \in C^2\}$ is called a fully quadratic class of models if*

1. *There exist positive constants κ_{ef}, κ_{eg}, and κ_{eh} 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 Hessian, and such that*

   - *the error between the Hessian of the model and the Hessian of the function satisfies*

   $$\|\nabla^2 f(x+s) - \nabla^2 m(x+s)\| \ \leq \ \kappa_{eh}\Delta, \quad \forall s \in B(0;\Delta), \tag{2.21}$$

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

   $$\|\nabla f(x+s) - \nabla m(x+s)\| \ \leq \ \kappa_{eg}\Delta^2, \quad \forall s \in B(0;\Delta), \tag{2.22}$$

   and

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

   $$|f(x+s) - m(x+s)| \ \leq \ \kappa_{ef}\Delta^3, \quad \forall s \in B(0;\Delta). \tag{2.23}$$

   Such a model m is called fully quadratic 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 quadratic on $B(x;\Delta)$ (we will say that a certificate has been provided and the model is certifiably fully quadratic),*

- *or find a model $\tilde{m} \in M$ that is fully quadratic on $B(x;\Delta)$.*

As in the linear case, model-improvement algorithms are based on Lagrange polynomials or factorizations of the interpolation matrices (see [23, Chapter 6]). Below, we review only the form of the bounds (2.21), (2.22), and (2.23) for quadratic interpolation and regression.

Quadratic interpolation models

We now consider the case where $d = 2$ ($m(x)$ is quadratic) and the number of points is equal to the number of basis functions, i.e., $p+1 = q+1$. We first state the required smoothness for f.

Assumption 2.2.4 *Assume that the function f is twice continuously differentiable in an open domain Ω containing $B(y^0;\Delta(Y))$ and $\nabla^2 f$ is Lipschitz continuous in Ω with constant $L_{\nabla^2 f} > 0$.*

Let us consider the matrix $\hat{Q}$ formed by the last p rows and columns of the scaled matrix $\hat{M}$ given in (2.16). The next theorem (see [23, Theorem 3.16]) says that we can build models satisfying the requirements (2.21), (2.22), and (2.23) of fully quadratic models using quadratic interpolation in the determined case.

Theorem 2.2.5 *Let Assumption 2.2.4 hold and $d = 2$. Assume that the set Y is a poised set of sample points (in the determined sense) contained in the ball $B(y^0;\Delta(Y))$ of radius $\Delta = \Delta(Y)$. Then for all points in $B(y^0;\Delta(Y))$, we have that*

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \;\leq\; \left(3\sqrt{2}p^{\frac{1}{2}}L_{\nabla^2 f}\|\hat{Q}^{-1}\|/2\right)\Delta,$$

$$\|\nabla f(y) - \nabla m(y)\| \;\leq\; \left(3(1+\sqrt{2})p^{\frac{1}{2}}L_{\nabla^2 f}\|\hat{Q}^{-1}\|/2\right)\Delta^2,$$

$$|f(y) - m(y)| \;\leq\; \left((6+9\sqrt{2})p^{\frac{1}{2}}L_{\nabla^2 f}\|\hat{Q}^{-1}\|/4 + L_{\nabla^2 f}/6\right)\Delta^3.$$

Quadratic regression models

It is possible to derive similar bounds for the case when $d = 2$ and there are more points than basis components, i.e., $p+1 > q+1$. We have seen that in this case one can compute regression models as least-squares solutions of (2.14). For this purpose, let us consider the reduced singular value decomposition of the scaled matrix $\hat{M} = \hat{U}\hat{\Sigma}\hat{V}^\top$ given in (2.16). The next theorem (see [23, Theorem 4.13]) says that we can build models satisfying the requirements (2.21), (2.22), and (2.23) of fully quadratic models using quadratic interpolation in the overdetermined or regression senses.

Theorem 2.2.6 *Let Assumption 2.2.4 hold and $d = 2$. Assume that the set Y is a poised set of sample points (in the regression sense) contained in the ball $B(y^0;\Delta(Y))$ of radius $\Delta = \Delta(Y)$. Then, for all points y in $B(y^0;\Delta(Y))$, we have*

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \;\leq\; \left(L_{\nabla^2 f} + \sqrt{2}\bar{p}^{\frac{1}{2}}L_{\nabla^2 f}/2\|\hat{\Sigma}^{-1}\|\right)\Delta,$$

$$\|\nabla f(y) - \nabla m(y)\| \;\leq\; \left(L_{\nabla^2 f} + (n^{\frac{1}{2}} + \sqrt{2}\bar{p}^{\frac{1}{2}})/2L_{\nabla^2 f}\|\hat{\Sigma}^{-1}\|\right)\Delta^2,$$

$$|f(y)-m(y)| \leq \left(L_{\nabla^2 f}/2+(1/2+n^{\frac{1}{2}}/2+\sqrt{2}\bar{p}^{\frac{1}{2}}/4)L_{\nabla^2 f}\|\hat{\Sigma}^{-1}\|\right)\Delta^3,$$

where $\bar{p}=n(n+1)/2$.

Finally, a natural question to pose is whether random sample sets can lead to fully linear or fully quadratic models in the framework of polynomial interpolation or regression. In the linear case, the question seems related to the condition number of random matrices. In the quadratic case, it was recently shown that, by randomly generating the points in the sample set following a uniform distribution, it is possible to build quadratic interpolating polynomials that are fully quadratic with high probability (see [5]). Such a technique can take advantage of the sparsity in the Hessian of the function to be modeled even without any prior knowledge of its sparsity pattern. For instance, it is proved in [5] that if the number of non-zero elements in the Hessian of the function is $\mathscr{O}(n)$, then random generation of $\mathscr{O}(n(logn)^4)$ sample points (instead of the $\mathscr{O}(n^2)$ required for the deterministic quadratic case) is sufficient to build fully quadratic models with high probability.

2.3 A derivative-free trust-region framework for smooth functions

Typically, a derivative-free trust-region algorithm starts with a chosen initial sample set, built around an initial starting point. Such a sample set can be chosen so that it has favorable geometrical properties. At each iteration, a quadratic model is built using the current sample set and then minimized inside the trust region. The approximate solution of the trust-region subproblem provides a step and thus a trial point. Whether the iteration is successful or not, this trial point can be included in the current sample set, possibly by removing a point from it. Such a scheme, or any variant one can think of, results in an iterative update of the sample set from which the models are built.

The quadratic model built around the current iterate x_k is now written as

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

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 trust region is a ball $B(x_k;\Delta_k)$ centered at x_k and of radius Δ_k. A major 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 same between f_k and $f(x_k)$). The matrix H_k is a good approximation for $\nabla^2 f(x_k)$ if the model is fully quadratic.

Given the accuracy properties of these models, one can then see that a number of modifications must be made in trust-region algorithms in the absence of derivatives. The modifications to a derivative based trust-region method (like Algorithm 2.1.1) are essentially three.

A first fundamental modification is that the trust-region radius should not be reduced in unsuccessful iterations unless the quality of the model is good. In fact, in the presence of derivatives or when the models are always fully linear, one knows that after a finite number of reductions of the trust-region radius, a successful iteration is generated and the method moves on (see, e.g., Lemma 2.1.2). Without using derivatives, when the ratio between the actual and the predicted decrease is not large enough, one should first promote a model-improvement before reducing the trust-region radius. Thus, and

this is the second fundamental modification, a model-improvement step must be included in each iteration. A third fundamental difference lies in the so-called criticality step. The algorithm should accept a step when the model predicts a good relative decrease in the objective function (since there is a cost at evaluating the objective function at the trial point). However, such successful iterations by themselves drive only the gradient of the model to zero, not necessarily the gradient of the objective function. One must then include a new step, called the criticality step, that ensures that when the model gradient is small, the models are fully linear in trust regions where the radius is of the order of the model gradient.

We describe below the derivative-free trust-region framework proposed and analyzed in [22] (and also described in the book [23]). As opposed to Algorithm 2.1.1, it contains already provision for accepting new iterates based on simple decrease of the objective function. Note that the incorporation of the criticality step complicates matters significantly. The authors in [22, 23] have chosen to work with incumbent models (a subject that will be revisited in Chapter 4 of this thesis) and thus the notation *icb* below. The following algorithm is (verbatim) Algorithm 4.1 in [22] (or Algorithm 10.1 in [23]). Note that at the end of the criticality step (Step 1 below) the trust-region is set as given in (2.24). This ensures that Δ_k is the number in $[\tilde{\Delta}_k, \Delta_k^{icb}]$ closest to $\beta\|\tilde{g}_k\|$. The outcome of the criticality step is a radius $\tilde{\Delta}_k \in (0, \mu\|\tilde{g}_k\|]$, where $\tilde{g}_k$ is the gradient of the latest model then computed. But $\tilde{\Delta}_k$ may be too small and so it is reset to $\beta\|\tilde{g}_k\|$ in that case (with $\mu > \beta > 0$). The update (2.24) also guarantees that the trust-region radius is not increased in the criticality step.

Algorithm 2.3.1 Derivative-free trust-region method using fully linear models

Step 0 (initialization): Choose a fully linear class of models M and a corresponding model-improvement algorithm (see, e.g., [20]). Choose an initial point x_0 and $\Delta_{max} > 0$. We assume that an initial model $m_0^{icb}(x_0 + s)$ (with gradient and possibly the Hessian at $s = 0$ given by g_0^{icb} and H_0^{icb}, respectively) and a trust-region radius $\Delta_0^{icb} \in (0, \Delta_{max}]$ are given.

The constants η_0, η_1, γ, γ_{inc} ε_c, β, μ, and α are also given and satisfy the conditions $0 \leq \eta_0 \leq \eta_1 < 1$ (with $\eta_1 \neq 0$), $0 < \gamma < 1 < \gamma_{inc}$, $\varepsilon_c > 0$, $\mu > \beta > 0$, and $\alpha \in (0, 1)$. Set $k = 0$.

Step 1 (criticality step): If $\|g_k^{icb}\| > \varepsilon_c$ then $m_k = m_k^{icb}$ and $\Delta_k = \Delta_k^{icb}$.

If $\|g_k^{icb}\| \leq \varepsilon_c$ then proceed as follows. Call the model-improvement algorithm to attempt to certify if the model m_k^{icb} is fully linear on $B(x_k; \Delta_k^{icb})$. If at least one of the following conditions holds,

- the model m_k^{icb} is not certifiably fully linear on $B(x_k; \Delta_k^{icb})$,
- $\Delta_k^{icb} > \mu\|g_k^{icb}\|$,

then apply Algorithm 2.3.2 (described below) to construct a model $\tilde{m}_k(x_k + s)$ (with gradient and possibly the Hessian at $s = 0$ given by $\tilde{g}_k$ and $\tilde{H}_k$, respectively), which is fully linear (for some constants κ_{ef}, κ_{eg}, and κ_{blg}, which remain the same for all iterations of Algorithm 2.3.1) on the ball $B(x_k; \tilde{\Delta}_k)$, for some $\tilde{\Delta}_k \in (0, \mu\|\tilde{g}_k\|]$ given by Algorithm 2.3.2. In such a case set

$$m_k = \tilde{m}_k \text{ and } \Delta_k = \min\{\max\{\tilde{\Delta}_k, \beta\|\tilde{g}_k\|\}, \Delta_k^{icb}\}. \tag{2.24}$$

Otherwise set $m_k = m_k^{icb}$ and $\Delta_k = \Delta_k^{icb}$.

Step 2 (step calculation): Compute a step s_k that sufficiently reduces the model m_k (in the sense of (2.2)) 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

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

If $\rho_k \geq \eta_1$ or if both $\rho_k \geq \eta_0$ and the model is fully linear (for the positive constants κ_{ef}, κ_{eg}, and κ_{blg}) on $B(x_k; \Delta_k)$, then $x_{k+1} = x_k + s_k$ and the model is updated to include the new iterate in the sample set, resulting in a new model $m^{icb}_{k+1}(x_{k+1} + s)$ (with gradient and possibly the Hessian at $s = 0$ given by g^{icb}_{k+1} and H^{icb}_{k+1}, respectively); otherwise the model and the iterate remain unchanged ($m^{icb}_{k+1} = m_k$ and $x_{k+1} = x_k$).

Step 4 (model improvement): If $\rho_k < \eta_1$ use the 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^{icb}_{k+1} to be the (possibly improved) model.

Step 5 (trust-region radius update): Set

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

Increment k by one and go to Step 1.

It is was proved in [22, Lemma 3.2] (see also [23, Lemma 10.25]) that if a model is fully linear in a ball, so it is in any larger concentric one. Thus, the possible increase of the trust-region radius at the end of the criticality step does not pose problems to the convergence theory. Note also that since we do not state such a result formally in this thesis, we have omitted the reference to the Lipschitz constant of the gradient of the model in Definition 2.2.1.

The procedure called in the criticality step (Step 1 of Algorithm 2.3.1) is described in the following algorithm, also taken verbatim from Algorithm 4.2 in [22] (see also Algorithm 10.2 in [23]).

Algorithm 2.3.2 (Criticality step: first order) *This algorithm is only applied if $\|g^{icb}_k\| \leq \varepsilon_c$ and at least one of the following holds: the model m^{icb}_k is not certifiably fully linear on $B(x_k; \Delta^{icb}_k)$ or $\Delta^{icb}_k > \mu\|g^{icb}_k\|$. The constant $\alpha \in (0,1)$ is chosen at Step 0 of Algorithm 2.3.1.*

Initialization: Set $i = 0$. Set $m^{(0)}_k = m^{icb}_k$.

Repeat Increment i by one. Use the model-improvement algorithm to improve the previous model $m^{(i-1)}_k$ until it is fully linear on $B(x_k; \alpha^{i-1}\Delta^{icb}_k)$ (notice that this can be done in a finite, uniformly

bounded number of steps given the choice of the model-improvement algorithm in Step 0 of Algorithm 2.3.1). Denote the new model by $m_k^{(i)}$. Set $\tilde{\Delta}_k = \alpha^{i-1}\Delta_k^{icb}$ and $\tilde{m}_k = m_k^{(i)}$.

Until $\tilde{\Delta}_k \leq \mu\|g_k^{(i)}\|$**.**

To better understand Algorithm 2.3.1, it helps enumerating the different types of iterations.

- successful iterations: $\rho_k \geq \eta_1$. This is when the new iterate is accepted. The trust-region radius in maintained or increase.
- acceptable iterations: $\eta_1 > \rho_k \geq \eta_0$ and the model m_k is fully linear. Here the new iterate is accepted and the trust-region is reduced.
- model-improving iterations: $\eta_1 > \rho_k$ and m_k is not certifiably fully linear. The model is improved. The new point is not accepted but might be included in the sample set. Importantly, the trust-region radius is not reduced (in fact it is kept the same).
- unsuccessful iteration: $\rho_k < \eta_0$ and the model m_k is fully linear. No acceptable decrease was obtained but because the model is accurate the trust-region radius can be reduced (as in derivative-based methods).

Global convergence to first-order stationary points

We will now describe the main first-order global convergence properties of Algorithm 2.3.1. This global convergence theory has been published in [22] and is also covered in [23]; the references will be to [22]. For this purpose f has to satisfy the same assumptions as in the first part of Section 2.1 of this thesis, but with the level set $L(x_0)$ there replaced now by $L_{enl}(x_0)$ in (2.5) for the purpose of smoothness. The assumptions are thus Assumption 2.1.1 and 2.2.1.

It is also required that the Hessian models are bounded (Assumption 2.1.3) and, implicitly, that there exists a fully linear class of models as in Definition 2.2.1.

The first result guarantees that the criticality step is well defined (see [22, Lemma 5.1]).

Lemma 2.3.1 *If $\nabla f(x_k) \neq 0$, Step 1 of Algorithm 2.3.1 will terminate in a finite number of improvement steps (by applying Algorithm 2.3.2).*

The next result is similar to Lemma 2.1.2 for derivative-based methods. It implies that a successful iteration will be achieved in a finite number of reductions of the trust-region radius. The result was stated as Lemma 5.2 in [22]. As model-improvement algorithms require also a finite number of steps to produce a fully linear model, one can say that a successful iteration is achieved in a finite number of iterations.

Lemma 2.3.2 *If m_k is fully linear on $B(x_k;\Delta_k)$ and*

$$\Delta_k \leq \min\left[\frac{1}{\kappa_{bhm}}, \frac{\kappa_{fcd}(1-\eta_1)}{4\kappa_{ef}}\right]\|g_k\|,$$

then the k-th iteration is successful.

The convergence theory progresses in [22, Lemma 5.4] by showing that $\nabla f(x_k)$ converges to zero if the number of successful iterations is finite. Then it is shown in [22, Lemma 5.5] that the trust-region radius converges to zero. This result is important by itself and thus stated below. It provides a stoping criteria for the algorithm in the absence of derivatives. The fact that $\|g_k\| \geq \min\{\varepsilon_c, \Delta_k/\mu\}$ is essential to derive this result and we see also here the relevance of the criticality step.

Lemma 2.3.3 *The trust-region radius converges to zero:*

$$\lim_{k\to+\infty} \Delta_k = 0.$$

Now we state the results that show that Algorithm 2.3.1 converges to stationary points. It can be proved that there is a subsequence of iterations that drives the model gradient to zero ([22, Lemma 5.6]) and that for any subsequence that drives the model gradient to zero the gradient of the function f also goes to zero ([22, Lemma 5.7]). From then, it is easily proved in [22] that the gradient goes to zero for a subsequence of iterates (Theorem 5.8).

Theorem 2.3.1 *Let Assumptions 2.1.1, 2.2.1, and 2.1.3 hold. Then,*

$$\liminf_{k\to+\infty} \nabla f(x_k) = 0.$$

Finally, it was also guaranteed in [22] that the whole sequence of gradients of f converges to zero (Theorem 5.9), following identical arguments as in the derivative-based case.

Theorem 2.3.2 *Let Assumptions 2.1.1, 2.2.1, and 2.1.3 hold. Then,*

$$\lim_{k\to+\infty} \nabla f(x_k) = 0.$$

Modifications to ensure global convergence to second-order stationary points

In part, the adaptation of Algorithm 2.3.1 to the second-order case follows what is done in the derivative-based case. As in Section 2.1, we define $\sigma_k = \sigma(x_k)$ as the second-order measure of criticality (2.8). Since now the models are not derivative based (and thus $g_k \neq \nabla f(x_k)$ and $H_k \neq \nabla^2 f(x_k)$) one defines the second-order measure of model criticality as

$$\sigma_k^m = \max\{\|g_k\|, -\lambda_{\min}(H_k)\}. \tag{2.25}$$

The modifications to Algorithm 2.3.1 to make it globally convergent to second-order criticality points consists, essentially, of replacing $\|g_k\|$ by σ_k^m and the use of fully linear models by the use of fully quadratic models. Another relevant modification that must be made in Algorithm 2.3.1, if one wants to achieve a lim-type result in the second-order case, concerns (as we mentioned in Section 2.1) the update of the trust-region radius for successful iterations (when ρ_k is sufficiently large). Here, instead of appealing to the known fix of increasing the trust-region radius for all such iterations, we follow the presentation in [22] and increase the trust-region radius only when it is small compared to the (model) measure of stationarity σ_k^m.

Algorithm 2.3.3 Derivative-free trust-region method using fully quadratic models

Step 0 (initialization): Choose a fully quadratic class of models M and a corresponding model-improvement algorithm (see, e.g., [20]). Choose an initial point x_0 and $\Delta_{max} > 0$. We assume that an initial model $m_0^{icb}(x_0+s)$ (with gradient and Hessian at $s=0$ given by $g_0^{icb} = \nabla f(x_0)$ and $H_0^{icb} = \nabla^2 f(x_0)$, respectively), with $\sigma_0^{m,icb} = \max\{\|g_0^{icb}\|, -\lambda_{\min}(H_0^{icb})\}$, and a trust-region radius $\Delta_0^{icb} \in (0, \Delta_{max}]$ are given.

The constants η_0, η_1, γ, γ_{inc} ε_c, β, μ, and α are also given and satisfy the conditions $0 \leq \eta_0 \leq \eta_1 < 1$ (with $\eta_1 \neq 0$), $0 < \gamma < 1 < \gamma_{inc}$, $\varepsilon_c > 0$, $\mu > \beta > 0$, and $\alpha \in (0,1)$. Set $k = 0$.

Step 1 (criticality step): If $\sigma_k^{m,icb} > \varepsilon_c$ then $m_k = m_k^{icb}$ and $\Delta_k = \Delta_k^{icb}$.

If $\sigma_k^{m,icb} \leq \varepsilon_c$ then proceed as follows. Call the model-improvement algorithm to attempt to certify if the model m_k^{icb} is fully quadratic on $B(x_k; \Delta_k^{icb})$. If at least one of the following conditions holds,

- the model m_k^{icb} is not certifiably fully quadratic on $B(x_k; \Delta_k^{icb})$,
- $\Delta_k^{icb} > \mu \sigma_k^{m,icb}$,

then apply Algorithm 2.3.4 (described below) to construct a model $\tilde{m}_k(x_k+s)$ (with gradient and Hessian at $s=0$ given by $\tilde{g}_k$ and $\tilde{H}_k$, respectively), with $\tilde{\sigma}_k^m = \max\{\|\tilde{g}_k\|, -\lambda_{\min}(\tilde{H}_k)\}$, which is fully quadratic (for some constants κ_{ef}, κ_{eg}, κ_{eh}, and κ_{blh}, which remain the same for all iterations of Algorithm 2.3.3) on the ball $B(x_k; \tilde{\Delta}_k)$ for some $\tilde{\Delta}_k \in (0, \mu\tilde{\sigma}_k^m]$ given by Algorithm 2.3.4. In such a case set

$$m_k = \tilde{m}_k \text{ and } \Delta_k = \min\{\max\{\tilde{\Delta}_k, \beta\tilde{\sigma}_k^m\}, \Delta_k^{icb}\} \tag{2.26}$$

Note that Δ_k is selected to be the number in $[\tilde{\Delta}_k, \Delta_k^{icb}]$ closest to $\beta\|\tilde{\sigma}_k^m\|$. Otherwise set $m_k = m_k^{icb}$ and $\Delta_k = \Delta_k^{icb}$.

Step 2 (step calculation): Compute a step s_k that sufficiently reduces the model m_k (in the sense of (2.7)) 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

$$\rho_k = \frac{f(x_k) - f(x_k+s_k)}{m_k(x_k) - m_k(x_k+s_k)}.$$

If $\rho_k \geq \eta_1$ or if both $\rho_k \geq \eta_0$ and the model is fully quadratic (for the positive constants κ_{ef}, κ_{eg}, κ_{eh}, and κ_{blh}) on $B(x_k; \Delta_k)$, then $x_{k+1} = x_k + s_k$ and the model is updated to include the new iterate in the sample set, resulting in a new model $m_{k+1}^{icb}(x_{k+1}+s)$ (with gradient and Hessian at $s=0$ given by g_{k+1}^{icb} and H_{k+1}^{icb}, respectively), with $\sigma_{k+1}^{m,icb} = \max\{\|g_{k+1}^{icb}\|, -\lambda_{\min}(H_{k+1}^{icb})\}$; otherwise the model and the iterate remain unchanged ($m_{k+1}^{icb} = m_k$ and $x_{k+1} = x_k$).

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

- attempt to certify that m_k is fully quadratic on $B(x_k; \Delta_k)$,

- if such a certificate is not obtained, we say that m_k is not certifiably fully quadratic and make one or more suitable improvement steps.

Define m_{k+1}^{icb} to be the (possibly improved) model.

Step 5 (trust-region radius update): Set

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

Increment k by one and go to Step 1.

Now we describe the second-order version of the criticality step. Note that at the end of the criticality step (Step 1 above) the trust-region is set as given in (2.26). This ensures that Δ_k is the number in $[\tilde{\Delta}_k, \Delta_k^{icb}]$ closest to $\beta\|\tilde{\sigma}_k^m\|$. The outcome of the criticality step is a radius $\tilde{\Delta}_k \in (0, \mu\|\tilde{\sigma}_k^m\|]$, where $\tilde{\sigma}_k^m$ is the second-order measure of criticality of the latest model then computed. But $\tilde{\Delta}_k$ may be too small and so it is reset to $\beta\|\tilde{\sigma}_k^m\|$ in that case (with $\mu > \beta > 0$). The update (2.26) also guarantees that the trust-region radius is not increased in the criticality step.

Algorithm 2.3.4 (Criticality step: second order) *This algorithm is only applied if $\sigma_k^{m,icb} \leq \varepsilon_c$ and at least one the following holds: the model m_k^{icb} is not certifiably fully quadratic on $B(x_k; \Delta_k^{icb})$ or $\Delta_k^{icb} > \mu\sigma_k^{m,icb}$. The constant $\alpha \in (0,1)$ is chosen at Step 0 of Algorithm 2.3.3.*

Initialization: Set $i = 0$. Set $m_k^{(0)} = m_k^{icb}$.

Repeat Increment i by one. Improve the previous model $m_k^{(i-1)}$ until it is fully quadratic on $B(x_k; \alpha^{i-1}\Delta_k^{icb})$ (notice that this can be done in a finite, uniformly bounded number of steps, given the choice of the model-improvement algorithm in Step 0 of Algorithm 2.3.3). Denote the new model by $m_k^{(i)}$. Set $\tilde{\Delta}_k = \alpha^{i-1}\Delta_k^{icb}$ and $\tilde{m}_k = m_k^{(i)}$.

Until $\tilde{\Delta}_k \leq \mu(\sigma_k^m)^{(i)}$.

There is a one-to-one correspondence between the results of the first-order and second-order cases and so we will list them next without introducing them first. The results are taken from [22]: Lemma 2.3.4 is Lemma 7.3 in [22]; Lemma 2.3.5 is Lemma 7.4; Lemma 2.3.6 is Lemma 7.7; Theorem 2.3.3 is Theorem 7.10, and Theorem 2.3.4 is Theorem 7.11. As for the assumptions they are Assumptions 2.1.1 and 2.2.3.

Lemma 2.3.4 *If $\sigma(x_k) \neq 0$, Step 1 of the Algorithm 2.3.3 will terminate in a finite number of improvement steps (by applying Algorithm 2.3.4).*

Lemma 2.3.5 *If m_k is fully quadratic on $B(x_k; \Delta_k)$ and*

$$\Delta_k \leq \min\left[\frac{1}{\kappa_{bhm}}, \frac{\kappa_{fod}(1-\eta_1)}{4\kappa_{ef}\Delta_{max}}, \frac{\kappa_{fod}(1-\eta_1)}{4\kappa_{ef}}\right]\sigma_k^m,$$

then the k-th iteration is successful.

Lemma 2.3.6 *The trust-region radius converges to zero:*

$$\lim_{k\to+\infty} \Delta_k = 0.$$

Theorem 2.3.3 *Let Assumptions 2.1.1, 2.2.3, and 2.1.3 hold. Then,*

$$\liminf_{k\to+\infty} \sigma(x_k) = 0.$$

Theorem 2.3.4 *Let Assumptions 2.1.1, 2.2.3, and 2.1.3 hold. Then,*

$$\lim_{k\to+\infty} \sigma(x_k) = 0.$$

In the proof of Theorem 2.3.4 (Theorem 7.11 in [22]), there is something significantly different from the first-order case as the scheme to update the trust-region radius has been changed in successful iterations, as we commented earlier.

2.4 Other derivative-free model-based approaches

The derivative-free trust-region framework of [22] described in Section 2.3 has been used by several authors. Billups, Larson, and Graf [8] first extended the theory developed for quadratic polynomial regression in the book [23, Chapter 4] from the regression case to the weighted regression case, including the definitions of Lagrange polynomials, basic properties, and error bounds. Then, based on the material of [22] and [23, Chapter 6], they developed a scheme to produce or update sets with good geometrical properties for weighted quadratic regression. Lastly, their trust-region framework is based on [22] thus exhibiting the same global convergence properties. Wild, Regis, and Shoemaker [66] proposed the use of radial basis function interpolation models in the framework [22] (see Section 2.3). Given the non-linearity of such basis functions, their fully linear models are non-linear, of non-quadratic type. As in Lagrange polynomials, each radial basis function is associated with a sample point, being of the form $\phi(\|x-y^i\|)$ for a given sample point y^i and a certain real function ϕ (and thus constant for any sphere centered at y^i). They characterized the types of radial basis functions that fit the fully linear requirements and, using the framework [22] (see Section 2.3), showed global convergence of the resulting algorithms to first-order stationary points. Perhaps due to the known ability of radial basis functions to approximate curvature function, they reported good numerical performance for their trust-region approach.

The influence of the geometry of the sample sets used for for interpolation/regression on the performance of the corresponding derivative-free trust-region algorithms has always been a relevant question. We have seen in this chapter that such geometry has to be good to built models that match the order of accuracy of the corresponding Taylor ones. But can an algorithm perform relatively well totally ignoring the geometry of the sample sets? Fasano, Morales, and Nocedal [32] considered a numerical setting where a determined quadratic interpolation model is built at the first iteration from $(n+1)(n+2)/2$ points and the sample set is kept with the same cardinality along the iterations.

Essentially, at each iteration, they bring into the sample set the new trial point and discard the sample point farthest away from the current iterate. Although the condition number of the interpolation matrix was observed relatively high, they reported good numerical performance of the resulting trust-region method. Scheinberg and Toint [63] provided an example showing, however, that an approach that does not incorporate a criticality type step may converge to a non-stationary point — thus geometry cannot be totally ignored. They then suggested a geometry-improving step only when the model gradient is small. Global convergence for their method is the result of a self-correction property inherent to the combination of trust regions and polynomial interpolation models.

In [6], the authors proposed an analyzed probabilistic trust-region methods, based on sample sets where the points are randomly generated. At each iteration, the models are considered fully linear or fully quadratic with a certain favorable probability conditioned to the past iteration history. Convergence to first and second-order stationary points is proved but almost surely, i.e., with probability one. Later, in [40], it was proved that this methodology exhibits, with overwhelming probability, the same global rate of convergence as the the gradient method (a subject that will be covered in the next chapter).

The use of interpolation and regression models is not restricted to trust-region methods. Direct-search methods (see [46] and [23, Chapter 7 and 8] for a review) is another important class of rigorous derivative-free algorithms where these models have been used. Direct-search methods can be based on simplex sets (like the Nelder-Mead method [51]) but most of the existing ones are directional methods. Their formulation incorporates typically two steps, a search step and a poll step. The poll step is what determines the convergence properties of these methods and where directions are used. The search step is optional for convergence and is used in practice to improve the numerical performance. This search-poll framework was first introduced in [9]. The authors considered there a search step where (surrogate) models are managed or calibrated, and showed that such a step does not interfere in the global convergence properties to first-order stationarity of the underlying direct-search methods strictly based on polling. Using the concept of a search step, interpolation and regression models were then brought to direct search in [25] using quadratics. The idea is simple and consisted of collecting the points previously evaluated in the poll step. After a while there are enough points to build models with relevant curvature information. At each iteration, a model can be minimized in a region of interest to define a new iterate if significant decrease is achieved. If not, the direct-search iteration reverts to the (directional) poll step. The models were of minimum Frobenius type, when the number of previously evaluated points was below $(n+1)(n+2)/2$, and regression was considered when there were more than this number. The numerical experiments showed a great improvement over the original direct-search algorithm used. A similar approach was taken in [3] but using models built by radial basis functions. Note that trust-region methods can themselves incorporate a search step at the beginning of their iterations [42] although its use is not so relevant as in direct search since such methods already contemplate modeling.

Up to now we only discussed trust-region methods for unconstrained minimization. In a feasible method for constrained optimization, where all iterates satisfy the constraints, the geometry of the nearby constraints influence then the quality of the models. For instance, when using an active-set type approach in trust-region methods for derivative-free bound constrained optimization, a difficulty is that the set of interpolation points may get aligned at one or more active bounds and deteriorate

the quality of the interpolation set. Such difficulty is overcome in [38], where an active-set strategy method is developed by minimizing in the subspace of the free (non-active) variables, saving function evaluations from optimization in lower dimensional subspaces (the respective code is called BC-DFO). Other strategies have been developed by including all the constraints in the trust-region subproblems. This type of trust-region methods were implemented in the codes BOBYQA [60] (a generalization of NEWUOA [59] for bound constrained optimization) and DFO [DFO] (which also considers feasible regions defined by continuously differentiable functions for which gradients can be computed). Recently, extensions to linearly constrained problems have been provided in the codes LINCOA [61] and LCOBYQA [43]. Sampaio and Toint [47] introduced the DEFT-FUNNEL algorithm, and adaptation of the trust-funnel method presented in [35] to solve an equality-constrained nonlinear optimization problem. Their algorithm also includes self-correcting geometry steps.

Chapter 3

Worst case complexity of algorithms for continuous nonlinear optimization

For many decades the community of Nonlinear Optimization has analyzed their algorithms by establishing asymptotic global convergence results (in form of limits of measures of stationarity, as described in Sections 2.1 and 2.3 of this thesis) and by establishing fast local rates of convergence (like superlinear and quadratic) close to strict local minimizers. More recently, following a trend in the community of Convex Optimization, researchers have started to pay more attention to the worst case complexity (WCC) of nonlinear optimization algorithms and to their companion global rates (where no assumption is made about the starting point which may be far away from stationarity or local minimization). Most of these results concern the unconstrained optimization of smooth functions. Even more recently, the same trend has been followed in derivative-free optimization. In this chapter, we will review the main WCC and global rates obtained so far for the derivative-based and derivative-free cases.

3.1 WCC for optimization with derivatives

The first global rate or WCC bound for nonlinear optimization has been established by Nesterov [52] for the gradient or steepest descent method. For a better understanding of what is at stake, let us review in detail the method and the corresponding result.

Trust-region methods are globalization schemes where first a maximum size for a step is specified and then the step itself is computed. There is another large class of globalization schemes called line-search methods, where first a (descent) direction is calculated and then a step size (along such a direction) is determined (see, e.g., [55, Chapter 3]). The iterates in a line-search method are updated as $x_{k+1} = x_k + \alpha_k p_k$, where p_k is a descent direction and α_k is the step size. Assume now that the function is continuously differentiable. The gradient method (also known as the steepest-descent method) is a particular line-search method, where $p_k = -\nabla f(x_k)$ and thus it generates points using

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k). \tag{3.1}$$

The critical issue is the choice of the step length α_k. One typically considers step sizes satisfying certain conditions known as the Wolfe conditions.

The first Wolfe condition is the sufficient decrease condition (also known as the Armijo condition), given by

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^\top p_k, \tag{3.2}$$

for some $c_1 \in (0,1)$. It basically imposes a certain sufficient decrease on the objective function along p_k. We have already seen something similar in trust-region methods. In fact, in trust regions one imposes a condition of the form $\rho_k \geq \eta$, with $\eta > 0$. Given the definition of ρ_k in (2.4), $\rho_k \geq \eta$ is equivalent to $f(x_k) - f(x_k + s_k) \geq \eta[m_k(x_k) - m_k(x_k + s_k)]$. Thus, when the model m_k is linear ($H_k = 0$), one has $m_k(x_k) - m_k(x_k + s_k) = -\nabla f(x_k)^\top s_k$, and given that s_k in trust regions corresponds to $\alpha_k p_k$ in line searches, one can see that $\rho_k \geq \eta$ corresponds to the sufficient decrease condition (3.2) of line searches. When $p_k = -\nabla f(x_k)$, condition (3.2) becomes

$$f(x_k + \alpha_k p_k) \leq f(x_k) - c_1 \alpha_k \|\nabla f(x_k)\|^2. \tag{3.3}$$

The second Wolfe condition is the curvature condition, and it is imposed to avoid taking very small steps. In general it takes the form

$$\nabla f(x_k + \alpha_k p_k)^\top p_k \geq c_2 \nabla f(x_k)^\top p_k,$$

for some $c_2 \in (c_1, 1)$. For the gradient method, where $p_k = -\nabla f(x_k)$, it becomes

$$\nabla f(x_k + \alpha_k p_k)^\top p_k \geq -c_2 \|\nabla f(x_k)\|^2. \tag{3.4}$$

The gradient algorithm can be then stated as follows:

Algorithm 3.1.1 Gradient method (Wolfe conditions)

Choose initial point x_0. Let c_1 and c_2 be such that $0 < c_1 < c_2 < 1$.

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

Compute α_k satisfying (3.3) and (3.4).

Compute x_{k+1} as in (3.1).

We will now assume that the gradient of f is Lipschitz continuous (with Lipschitz constant $L_{\nabla f}$), as in Assumption 2.1.2. Under such an assumption and boundedness from below of f (Assumption 2.1.1), it is possible to prove the existence of intervals satisfying the Wolfe conditions ([55, Lemma 3.1]), which shows that Algorithm 3.1.1 is in some form well defined.

It is also possible to prove (see [55, Proof of Theorem 3.2]) that the iterates generated by Algorithm 3.1.1 satisfy

$$f(x_k) - f(x_{k+1}) \geq C_3 \|\nabla f(x_k)\|^2, \tag{3.5}$$

with

$$C_3 = c_1 \frac{1 - c_2}{L_{\nabla f}}.$$

This is the key condition used to prove the WCC bound and the associated global rate of convergence of Algorithm 3.1.1 ([52, Pages 29–32]):

Theorem 3.1.1 *Let Assumptions 2.1.1 and 2.1.2 hold. Then, the gradient method (Algorithm 3.1.1) generates a sequence of iterates* $\{x_k\}$ *such that*

$$\min_{0\leq j\leq k} \|\nabla f(x_j)\| \;\leq\; \left(\frac{f(x_0)-f_{low}}{C_3(k+1)}\right)^{1/2},$$

where $C_3 = c_1(1-c_2)/L_{\nabla f}$ *and, recall,* $L_{\nabla f}$ *is the Lipschitz constant for* ∇f *and* f_{low} *is a lower bound for* f.

Let $\varepsilon \in (0,1)$. *The gradient method takes at most*

$$\left\lceil C_3(f(x_0)-f_{low})\frac{1}{\varepsilon^2}-1\right\rceil$$

iterations to compute a x_k *such that* $\|\nabla f(x_k)\| \leq \varepsilon$.

One sees that the gradient decays at a sublinear rate $1/\sqrt{k}$ or, equivalently, that the WCC effort is of the order of ε^{-2}. No assumption on x_0 is made — the result is global.

In practical situations, it may be difficult to impose both Wolfe conditions at the same time. However, the effect of the imposition of the curvature condition (avoid small steps) can be achieved by imposing the sufficient decrease condition by means of a backtracking procedure. At each iteration, one starts by checking (3.3) for a certain constant value of α (below equal to b) and start reducing it (below by a factor of 2) until it is satisfied.

Algorithm 3.1.2 Gradient method (backtracking)

Choose initial point x_0. Let $c_1 \in (0,1)$ and $b > 0$.

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

Let α_k be the first α in $b, b/2, b/4, \ldots$ such that (3.3) is satisfied.

Compute x_{k+1} as in (3.1).

Under Assumptions 2.1.1 and 2.1.2, it is possible to show that Algorithm 3.1.2 is well defined in the sense that it is always possible to find α_k of the form given in the algorithm such that (3.3) is satisfied. Moreover, each iteration of Algorithm 3.1.2 satisfies (3.5) but now with

$$C_3 \;=\; c_1 \max\left(\frac{1-c_1}{L_{\nabla f}}, b\right).$$

Thus, Theorem 3.1.1 also holds for the gradient method with backtracking (Algorithm 3.1.2) under a simple redefinition of C_3.

In conclusion, as we have seen in Theorem 3.1.1, a gradient method based on the Wolfe conditions takes at most $\mathcal{O}(\varepsilon^{-2})$ iterations to drive the norm of the gradient of the objective function below $\varepsilon \in (0,1)$. The gradient decays at a sublinear rate of $1/\sqrt{k}$, independently of the starting point. The

result is also valid if only the sufficient decrease condition is imposed at each iteration by means of a backtracking procedure.

Cartis, Gould, and Toint [11] have proved that the bound $\mathcal{O}(\varepsilon^{-2})$ is sharp for the gradient method. The sharpness or tightness of the bound was established by constructing an example (for $n = 1$), dependent on an arbitrarily small parameter $\tau > 0$, for which the gradient method (using backtracking until sufficient decrease is satisfied as in Algorithm 3.1.2) requires, for any $\varepsilon \in (0,1)$, at least $\mathcal{O}(\varepsilon^{-2+\tau})$ iterations to reduce the norm of the gradient below ε. Interestingly, they have also shown a similar result for Newton's method.

Not surprisingly, a similar WCC bound of $\mathcal{O}(\varepsilon^{-2})$ has been proved by Gratton, Sartenaer, and Toint [41] for trust-region methods (where sufficient decrease is imposed).

There is, however, a class of methods for nonlinear optimization that exhibit better WCC: cubic regularization methods. The bound reduces to $\mathcal{O}(\varepsilon^{-1.5})$ iterations. The gradient decays globally at a rate of $1/k^{2/3}$. The method and its complexity were first introduced by Nesterov and Polyak [54] by using a cubic model where a multiple of a term of the form $\sigma\|s\|^3$ is added to the quadratic model. Here, σ is a positive parameter that plays the role of the trust-region radius in the sense of restricting or regularizing the norm of the step. Cartis, Gould, and Toint [10] introduced later an adaptive scheme for the cubic regularization parameter that generalizes the original results of [54] for more practical scenarios where the subproblems are solved inexactly and only first-order derivatives are used. The resulting approach has been coined adaptive cubic overestimation algorithm.

Very recently, Curtis, Robinson, and Samadi [24] modified the step acceptance criterion and the trust-region radius update of trust-region methods and were apparently able to prove that the resulting algorithm can also compute a gradient of norm smaller than ε in at most $\mathcal{O}(\varepsilon^{-1.5})$ iterations.

3.2 WCC for optimization without derivatives

The first global rate or WCC bound for nonlinear optimization without derivatives has been established by Vicente [64] for a class of direct-search methods. In fact, 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. Let us start also by reviewing these methods in a simple setting and look at the corresponding result. In the derivative-free case, we are not only interested in bounding the number of iterations but more importantly the number of function evaluations.

A directional method for minimizing a smooth function without the knowledge of the gradient must use (at each iteration) a set of directions which contains a descent one. Such a property is given by a positive spanning set (PSS), i.e., a set of vectors whose positive span is $\mathbb{R}^n$ (and by positive span one means the set of linear combinations using nonnegative scalars). The theory of positive spanning has been developed by Davis [26] and updated summaries for optimization can be found in [23, 46]. Given any PSS and any nonzero vector $w \in \mathbb{R}^n$, there exists a vector v in the PSS such that $w^\top v > 0$. When f is continuously differentiable, this implies that, as long as $\nabla f(x) \neq 0$, there is always a descent direction for f at x in any PSS (since there will always be a direction v in the PSS such that $-\nabla f(x)^\top v > 0$).

Direct-search methods form a class of methods for DFO characterized by updating the iterates based on the evaluation of the objective function on a finite number of points and without making any

attempt to build models or indirectly approximating derivatives. As we said in Section 2.4, they can be of directional or simplicial type (see [23]). The directional ones, when applied to smooth functions, are based on PSSs. As we also said in Section 2.4, such direct-search methods include two main steps, a search step and a poll step, but we will ignore here the search step as it is optional and does not interfere in the analysis of the algorithms. In the poll step, one evaluates the function at points of the form $x_k + \alpha_k d$, for directions d in a PSS, and where α_k is the step size. Not surprisingly, a new iterate must provide a sufficient decrease on the objective function, of the form $f(x_k + \alpha_k d_k) < f(x_k) - \alpha_k^2/2$, where $\alpha_k^2/2$ plays the role of the so-called *forcing function* [46]. Let us now state such a direct-search method (which is a simplified version of Algorithm 2.1 in [64]). The poll step can opportunistically move to the first point where sufficient decrease is found or can be complete (where all points are evaluated and the best is compared with the current one).

Algorithm 3.2.1 Direct-search method (polling)

Initialization: Choose a PSS D, an initial point x_0, and an initial step size $\alpha_0 > 0$. The constants $0 < \beta < 1 \leq \gamma$ are given. Set $k = 0$.

Step 1 (Poll Step): Order the set of poll points $P_k = \{x_k + \alpha_k d : d \in D\}$. Start evaluating f at the poll points following the chosen order. If a poll point $x_k + \alpha_k d$ is found such that $f(x_k + \alpha_k d_k) < f(x_k) - \alpha_k^2/2$, then set $x_{k+1} = x_k + \alpha_k d_k$ and declare the iteration successful. Otherwise, declare the iteration unsuccessful and set $x_{k+1} = x_k$.

Step 2 (Step size parameter update): If the iteration was successful, then maintain or increase the step size parameter: $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$. Otherwise, decrease the step size parameter $\alpha_{k+1} = \beta\alpha_k$. Increment k by one and go to Step 1.

It is possible to prove [46] that if the iteration k is unsuccessful, then

$$\|\nabla f(x_k)\| \leq \mathrm{cm}(D)^{-1} \left(L_{\nabla f} \frac{\max_{d \in D} \|d\|}{2} + \frac{1}{2 \min_{d \in D} \|d\|} \right) \alpha_k, \tag{3.6}$$

where $\mathrm{cm}(D)$ is the cosine measure of the PSS D, defined as

$$\mathrm{cm}(D) = \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|}.$$

The cosine measure of a PSS is always positive. For instance, the PSS $D_\oplus$ formed by the coordinate vectors and their negatives is such that $\mathrm{cm}(D_\oplus) = 1/\sqrt{n}$. The fact that (3.6) holds for unsuccessful iterations shows that the algorithm is well defined in the sense that a successful iteration will always be achieved in a finite number of reductions of the step size. It also provides a lower bound for the step size similar to what holds in gradient methods. Thus, the combination of the fact that (3.6) holds for unsuccessful iterations with the fact that a sufficient decrease condition, $f(x_k) - f(x_{k+1}) \geq \alpha_k^2/2$, is achieved in successful iterations leads to a complexity result (see [64, Corollary 3.1]) similar to gradient methods.

Theorem 3.2.1 *Let Assumptions 2.1.1 and 2.1.2 hold. Let $\varepsilon > 0$. Then, direct search (Algorithm 3.2.1) takes at most*

$$\lceil E_1 \varepsilon^{-2} + E_2 \rceil$$

iterations to compute a x_k such that $\|\nabla f(x_k)\| \leq \varepsilon$, where

$$E_1 = \left(1 - \log_\beta(\gamma)\right) \frac{f(x_{k_0}) - f_{low}}{0.5\beta^2 L_1^2} - \log_\beta(\exp(1)),$$

$$E_2 = \log_\beta\left(\frac{\beta L_1 \exp(1)}{\alpha_{k_0}}\right) + \frac{f(x_0) - f_{low}}{0.5\alpha_0^2},$$

$$L_1 = \min(1, L_2^{-1}),$$

$$L_2 = cm(D)^{-1}\left(L_{\nabla f}\frac{\max_{d\in D}\|d\|}{2} + \frac{1}{2\min_{d\in D}\|d\|}\right),$$

and k_0 is the index of the first unsuccessful iteration (and recall that $L_{\nabla f}$ is the Lipschitz constant for ∇f and f_{low} is a lower bound for f).

Moreover, $\{\min_{0\leq j\leq k}\|\nabla f(x_j)\|\}$ converges sublinearly to zero at the rate $1/\sqrt{k}$.

The above result is proved in [64] in a more general setting. One can let the algorithm choose different PSSs in different iterations as long as their cosine measures are bounded away from zero (meaning that they cannot arbitrarily loose their defining property) and their vectors are bounded below and above. One can consider a search step as long as it also imposes sufficient decrease.

From Theorem 3.2.1 it is also possible to deduce the WCC of direct search (like Algorithm 3.2.1) in terms of the number of function evaluations. In fact, one can see from Theorem 3.2.1 that E_1 is of the order of $1/\operatorname{cm}(D)^2$ and that E_2 does not depend on D. On the other hand, the maximum number of function evaluations at each iteration is the cardinal $|D|$. In the case where $D = D_\oplus$, one has $\operatorname{cm}(D) = 1/\sqrt{n}$ and $|D| = 2n$ and the following corollary is thus well posed (see [64, Corollary 3.2]).

Corollary 3.2.1 *Let all the assumptions of the Theorem 3.2.1 hold. Let D be a PSS such that $\operatorname{cm}(D) = \mathcal{O}(1/\sqrt{n})$ and $|D| = \mathcal{O}(n)$. To reduce the gradient below $\varepsilon \in (0,1)$, Algorithm 3.2.1 takes at most $\mathcal{O}(n^2\varepsilon^{-2})$ function evaluations.*

Recently it was proved in [29] that the factor n^2 is approximately optimal in this WCC bound, in the sense that no other PSS considered for Algorithm 3.2.1 would yield an order of n better than the one provided by $D_\oplus$. The result is the following [29, Theorem 4.1]:

Theorem 3.2.2 *There exists a universal constant $C_4 > 0$ such that*

$$\frac{|D|}{\operatorname{cm}(D)^2} \geq C_4 n^2,$$

for each $n \geq 1$ and each PSS D in $\mathbb{R}^n$.

In conclusion, as we have seen in Theorem 3.2.1, a direct-search method based on sufficient decrease takes at most $\mathcal{O}(\varepsilon^{-2})$ iterations and at most $\mathcal{O}(n^2\varepsilon^{-2})$ function evaluations to drive the norm

of the gradient of the objective function below $\varepsilon \in (0,1)$. The gradient decays also at a sublinear rate of $1/\sqrt{k}$, independently of the starting point. The fact that ε^{-2} is sharp in both bounds is shown in [64] by recasting direct search as a gradient method when $n = 1$ and then appealing to the example in [11]. The fact that n^2 is sharp is shown in [29] (see Theorem 3.2.2).

Cartis, Gould, and Toint [14] have derived a WCC bound of $\mathscr{O}(\varepsilon^{-1.5})$ for their derivative-free adaptive cubic overestimation algorithm in the smooth case, but using finite differences to approximate derivatives, matching the counterpart result for derivatives.

Many practical problems where derivatives are not available are non-smooth. Direct-search methods of directional type are particular tailored for dealing with non-smoothness as they do not fit models but rather explore directions. However, it becomes difficult to measure effort in the worst case without some type of smoothing or knowledge of the structure of non-smoothness. Garmanjani and Vicente [34], using a smoothing approach for direct search, have shown a WCC bound of $\mathscr{O}(|\log\varepsilon|\varepsilon^{-3})$ in the non-smooth case (where here ε refers to a threshold for the norm of the smoothing gradient and the smoothing parameter). We will return to this issue in more detail in Chapter 5 of this thesis, where we derive a similar result for smoothing trust-region methods. Similar WCC bounds were derived, in expectation, by Nesterov [53] for his random Gaussian smoothing approach.

The convex and strongly convex cases

Gradient and direct-search methods are faster (in theory) in the presence of convexity for smooth functions. Nesterov [52] proved that if the function is convex and the solution set is non-empty, the gradient method takes only $\mathscr{O}(\varepsilon^{-1})$ iterations to identify a point where the norm of the gradient is smaller that ε. It is also proved in [52] a sublinear global rate of $1/k$ for both $f(x_k) - f_*$ (where f_* is the optimal value) and $\|\nabla f(x_k)\|$. When the function is strongly convex, the WCC bound becomes $\mathscr{O}(-\log(\varepsilon))$ and the global rates becomes linear. Similar results were proved by Dodangeh and Vicente [28] for optimization without derivatives, using direct search based on sufficient decrease. The WCC bounds in terms of function evaluations are multiplied by a factor of n^2 as in the non-convex case.

Chapter 4

Worst case complexity of derivative-free trust-region methods

As reported in Sections 2.3–2.4, the global convergence properties of derivative-free trust-region methods are well studied, whether the convergence is for first-order or for second-order stationary points. In this chapter we will establish the worst-case complexity (WCC) analysis of such derivative-free trust-region methods (for the unconstrained minimization of smooth functions $f : \mathbb{R}^n \to \mathbb{R}$). First we will do it for first-order stationary points in Section 4.1 and then for second-order stationary points in Section 4.2. In both cases, due to the new way of looking at these algorithms introduced here, we will revisit some of their global convergence properties.

4.1 Complexity in determining first-order stationary points

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 [22] (see also [23, Algorithm 10.1] or Algorithm 2.3.1 in this thesis).

The algorithm revisited

The first modification concerns how the so-called criticality step is incorporated (see Algorithm 4.2 in [22], presentation in [23, Algorithm 10.2], or Algorithm 2.3.2 in this thesis). One knows from the counter example in [63] 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 thesis, 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 [22]; see Algorithm 2.3.1), which had to be used when the criticality step was invoked and changed the models coming from the previous iteration.

The second modification generalizes [22] 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 $\mathcal{O}(\varepsilon^{-3})$, but such a choice will be instrumental in the analysis of complexity of the smoothing trust-region approach of Section 5.3.

Algorithm 4.1.1 Derivative-free trust-region method using fully linear models (version WCC)

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 $\Delta_{k+1} = \max\{\gamma\Delta_k, \beta\|g_{k+1}\|\}$. 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 (2.3) 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 4.1.1 (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** ($\mathscr{C}^r$), taken at Step 1 and where the trust-region radius is reduced.
2. **Critical iterations** ($\mathscr{C}^{nr}$), taken at Step 1 and where the trust-region radius is not reduced.
3. **Successful iterations** ($\mathscr{S}$), 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** ($\mathscr{A}$), 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** ($\mathscr{U}$), 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** ($\mathscr{M}$), 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 $\mathscr{M}$ (or any iteration in $\mathscr{C}$) is of the order of n for a single function (see [23, Chapter 2] or the explanation about linear interpolation and regression models in Section 2.2).

For analyzing the algorithm, we gather all iterations that are not successful in $\mathscr{N} = \mathscr{C} \cup \mathscr{A} \cup \mathscr{U} \cup \mathscr{M}$, where $\mathscr{C} = \mathscr{C}^r \cup \mathscr{C}^{nr}$, and all iterations where Δ_k is reduced in $\mathscr{R} = \mathscr{C}^r \cup \mathscr{A} \cup \mathscr{U}$.

The two modifications described above do not restrict the general setting of [22]. However, a careful reader would notice that in [22] (see Algorithm 2.3.1 in Section 2.3) the criticality step is only applied when $\|g_k\| \leq \varepsilon_c$, with $\varepsilon_c > 0$. 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., $\varepsilon_c \geq \varepsilon$ 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.

Global convergence

Given that substantial modifications in the presentation of the algorithm are made relatively to the original description in [22] (Algorithm 2.3.1 in Section 2.3), it becomes necessary to redo the global convergence theory. Part of it would have to be done anyhow for the sole purpose of analyzing the worst case complexity.

As we have seen in Section 2.3, we need to assume that the objective function f is bounded from below (Assumption 2.1.1), that the model Hessians are uniformly bounded (Assumption 2.1.3), and that f has a Lipschitz continuous gradient in an open set containing initial enlarged level set $L_{enl}(x_0)$ (Assumption 2.2.1).

We will first show that the trust-region radius converges to zero. The proof is a modification of the proof of Lemma 5.5 in [22] (see also [23, Lemma 10.9]), which we have stated before as Lemma 2.3.6.

Lemma 4.1.1 *Let Assumptions 2.1.1 and 2.1.3 hold. Then*

$$\lim_{k\to+\infty} \Delta_k \;=\; 0.$$

Proof. First we assume that the number of successful iterations is finite. Suppose that the number of iterations in $\mathscr{R} = \mathscr{C}^r \cup \mathscr{A} \cup \mathscr{U}$ is also finite. Then we would have an infinite number of iterations either in $\mathscr{C}^{nr}$ or in $\mathscr{M}$. In the first case, a contradiction would be reached since after each iteration in $\mathscr{C}^{nr}$ (the last in a series of critical ones) the model is fully linear and we would either have an iteration in $\mathscr{S}$, $\mathscr{A}$, or in $\mathscr{U}$. 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 $\mathscr{C}^r$, $\mathscr{C}^{nr}$, $\mathscr{S}$, $\mathscr{A}$, or $\mathscr{U}$, which is not possible. Thus, there is an infinite number of iterations in $\mathscr{R} = \mathscr{C}^r \cup \mathscr{A} \cup \mathscr{U}$. 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 $\mathscr{S}$ is infinite. When k is in $\mathscr{S}$,

$$f(x_k) - f(x_{k+1}) \;\geq\; \eta_1 [m_k(x_k) - m_k(x_k + s_k)] + c_1 \Delta_k^p.$$

By using the bound on the fraction of Cauchy decrease (2.3) and Assumption 2.1.3, we have that

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

Since the iteration is not critical, $\|g_k\| \geq \Delta_k/\lambda$, and thus

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta_1 \kappa_{fcd}}{2\lambda} \Delta_k \min\left\{ \frac{\Delta_k}{\kappa_{bhm}\lambda}, \Delta_k \right\} + c_1 \Delta_k^p. \tag{4.1}$$

Given that $\mathscr{S}$ is considered infinite and f is assumed bounded from below, the right-hand side of (4.1) has to converge to zero for $k \in \mathscr{S}$. Hence $\lim_{k\in\mathscr{S}} \Delta_k = 0$, and the proof is completed when there are only successful iterations.

Now, if there exists an iteration k that is not successful then, due to the way in which the radii are updated at Step 5 of Algorithm 4.1.1, we have $\Delta_k \leq \gamma_{inc} \Delta_{s_k}$, where s_k is the last successful iteration before k. Since $\lim_{s_k\in\mathscr{S}} \Delta_{s_k} = 0$, the proof is completed in this case as well. ■

Having in mind the complexity results and the smoothing trust-region approach of Section 5.3, global convergence is established by proving now that the gradient of the objective function is of the order of the trust-region radius whenever this one is reduced.

Lemma 4.1.2 *Let Assumptions 2.1.3 and 2.2.1 hold. If k is an iteration for which Δ_k is reduced, then*

$$\|\nabla f(x_k)\| \;\leq\; C_5 \Delta_k + C_6 \Delta_k^{p-1},$$

where

$$C_5 = \kappa_{eg} + C_{0,1}, \quad C_{0,1} \;=\; \frac{1}{\min\left\{\beta, \frac{1}{\kappa_{bhm}}, \frac{\kappa_{fcd}(1-\eta_1)}{4\kappa_{ef}}\right\}}, \quad \textit{and} \quad C_6 = \frac{2c_1}{\kappa_{fcd}(1-\eta_1)}. \tag{4.2}$$

Proof. By assumption we have that $k \in \mathscr{R} = \mathscr{C}^r \cup \mathscr{A} \cup \mathscr{U}$. Let us suppose that $k \in \mathscr{A} \cup \mathscr{U}$. We will show first that

$$\|g_k\| \;\leq\; C_{0,1}\Delta_k + C_6\Delta_k^{p-1}. \tag{4.3}$$

Assume by contradiction that (4.3) is false. Given that $C_{0,1} \geq \frac{4\kappa_{ef}}{\kappa_{fcd}(1-\eta_1)}$, we then obtain

$$\|g_k\| \;>\; \frac{4\kappa_{ef}}{\kappa_{fcd}(1-\eta_1)}\Delta_k + \frac{2c_1}{\kappa_{fcd}(1-\eta_1)}\Delta_k^{p-1},$$

which we rewrite as

$$1-\eta_1 \;>\; \frac{2\kappa_{ef}\Delta_k}{\frac{\kappa_{fcd}}{2}\|g_k\|} + \frac{c_1\Delta_k^{p-1}}{\frac{\kappa_{fcd}}{2}\|g_k\|}. \tag{4.4}$$

On the other hand, using (2.3) and $C_{0,1} \geq \kappa_{bhm}$, one has

$$m_k(x_k) - m_k(x_k+s_k) \;\geq\; \frac{\kappa_{fcd}}{2}\|g_k\| \min\left\{\frac{\|g_k\|}{\kappa_{bhm}}, \Delta_k\right\} \;\geq\; \frac{\kappa_{fcd}}{2}\|g_k\|\Delta_k. \tag{4.5}$$

Hence, we have, from (4.4) and (4.5) and the fully linearity (2.12) of the model at both $s = 0$ and $s = s_k$,

$$\begin{aligned}
1-\eta_1 \;&\geq\; \left|\frac{f(x_k+s_k) - m_k(x_k+s_k)}{m_k(x_k) - m_k(x_k+s_k)}\right| + \left|\frac{f(x_k) - m_k(x_k)}{m_k(x_k) - m_k(x_k+s_k)}\right| \\
&\quad + \left|\frac{c_1\Delta_k^p}{m_k(x_k) - m_k(x_k+s_k)}\right| \\
&\geq\; \left|\frac{f(x_k) - f(x_k+s_k) - c_1\Delta_k^p - m_k(x_k) + m_k(x_k+s_k)}{m_k(x_k) - m_k(x_k+s_k)}\right| \\
&=\; \left|\frac{f(x_k) - f(x_k+s_k) - c_1\Delta_k^p}{m_k(x_k) - m_k(x_k+s_k)} - 1\right| \\
&=\; |\rho_k - 1|.
\end{aligned}$$

Therefore, we have $\rho_k > \eta_1$, implying that the iteration is successful and contradicting the fact that $k \in \mathscr{A} \cup \mathscr{U}$. We have thus proved (4.3) for $k \in \mathscr{A} \cup \mathscr{U}$. To establish the result of the lemma when $k \in \mathscr{A} \cup \mathscr{U}$, it remains to use (2.11) and write

$$\begin{aligned}
\|\nabla f(x_k)\| \;&\leq\; \|\nabla f(x_k) - g_k\| + \|g_k\| \;\leq\; \kappa_{eg}\Delta_k + C_{0,1}\Delta_k + C_6\Delta_k^{p-1} \\
&=\; C_5\Delta_k + C_6\Delta_k^{p-1}.
\end{aligned}$$

Let us now suppose that $k \in \mathscr{C}^r$. If k is not the last critical iteration in a series of them, then $\Delta_{k+1} = \gamma\Delta_k$ and $\Delta_{k+1} > \lambda\|g_{k+1}\|$. Thus,

$$\begin{aligned}\|\nabla f(x_k)\| &= \|\nabla f(x_{k+1})\| \leq \|\nabla f(x_{k+1}) - g_{k+1}\| + \|g_{k+1}\| \\ &\leq \kappa_{eg}\Delta_{k+1} + \|g_{k+1}\| \leq \kappa_{eg}\gamma\Delta_k + \frac{\gamma\Delta_k}{\lambda} \leq \left(\kappa_{eg} + \frac{1}{\beta}\right)\gamma\Delta_k \\ &\leq C_5\Delta_k + C_6\Delta_k^{p-1}.\end{aligned}$$

If k is the last critical iteration in a series of them, then due to $\Delta_{k+1} = \max\{\gamma\Delta_k, \beta\|g_{k+1}\|\}$, either $\Delta_{k+1} = \gamma\Delta_k$ or $\Delta_{k+1} = \beta\|g_{k+1}\| < \Delta_k$. 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,

$$\begin{aligned}\|\nabla f(x_k)\| &= \|\nabla f(x_{k+1})\| \leq \|\nabla f(x_{k+1}) - g_{k+1}\| + \|g_{k+1}\| \\ &\leq \kappa_{eg}\Delta_{k+1} + \frac{\Delta_{k+1}}{\beta} \leq \kappa_{eg}\Delta_k + \frac{\Delta_k}{\beta} = \left(\kappa_{eg} + \frac{1}{\beta}\right)\Delta_k \\ &\leq C_5\Delta_k + C_6\Delta_k^{p-1},\end{aligned}$$

and we establish the result of the lemma also for $k \in \mathscr{C}^r$. ■

A global convergence result follows directly from Lemma 4.1.2 and the asymptotic behavior of the trust-region radius.

Theorem 4.1.1 *Let Assumptions 2.1.1, 2.1.3, and 2.2.1 hold. Then*

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

Proof. By Lemma 4.1.1, there is an infinite subsequence of iterations where the trust-region radius is reduced, to which then we can apply Lemma 4.1.2. ■

Worst case complexity

In this section, we derive the worst-case complexity analysis of Algorithm 4.1.1. 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.

Lemma 4.1.3 *Let Assumptions 2.1.3 and 2.2.1 hold. Let $\varepsilon \in (0,1)$. Let k_0 be the first iteration where Δ_k is reduced. For every iteration $k \geq k_0$ of the algorithm, if $\|\nabla f(x_j)\| > \varepsilon$ for $j = k_0, \ldots, k$, then*

$$\Delta_k \geq \gamma C_7 \varepsilon^{\frac{1}{\min(p-1,1)}},$$

where

$$C_7 = \min\left(1, (C_5 + C_6)^{-\frac{1}{\min(p-1,1)}}\right), \tag{4.6}$$

with C_5 and C_6 given in (4.2).

Proof. Let $k \geq k_0$ be an iteration where Δ_k is reduced. When $\Delta_k < 1$, by applying Lemma 4.1.2,

$$\varepsilon \;<\; (C_5 + C_6)\max\{\Delta_k, \Delta_k^{p-1}\} \;\leq\; (C_5 + C_6)\Delta_k^{\min(p-1,1)}.$$

If $\Delta_k \geq 1$, then $\Delta_k \geq \varepsilon$. Hence, considering both cases of $\Delta_k < 1$ and $\Delta_k \geq 1$, and the fact that $\varepsilon < 1$, we have

$$\Delta_k \;\geq\; C_7 \varepsilon^{\frac{1}{\min(p-1,1)}}.$$

The lemma is proved for all iterations $k \in \mathscr{R}$ such that $k \geq k_0$.

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

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

Theorem 4.1.2 *Let Assumptions 2.1.1, 2.1.3, and 2.2.1 hold. Let k_0 be the index of the first iteration where Δ_k is reduced (which must exist from Lemma 4.1.1). Given any $\varepsilon \in (0,1)$, assume that $\|\nabla f(x_{k_0})\| > \varepsilon$ and let $\bar{k}$ be the first iteration after k_0 such that $\|\nabla f(x_{\bar{k}})\| \leq \varepsilon$. Then, to achieve $\|\nabla f(x_{\bar{k}})\| \leq \varepsilon$, starting from k_0, Algorithm 4.1.1 takes at most $|\mathscr{S}(k_0, \bar{k})|$ successful iterations, where*

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

where

$$L \;=\; \frac{\eta_1 \kappa_{fcd} \gamma^2 C_7^2}{2\lambda} \min\left\{\frac{1}{\kappa_{bhm}\lambda}, 1\right\} + c_1 \gamma^p C_7^p,$$

with C_7 given in (4.6) (and $\mathscr{S}(k_0, \bar{k})$ includes k_0 but excludes $\bar{k}$).

Proof. When $k \in \mathscr{S}$, using (2.3) and $\|g_k\| \geq \Delta_k/\lambda$, we have

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

Hence, by applying Lemma 4.1.3,

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta_1 \kappa_{fcd} \gamma^2 C_7^2}{2\lambda} \min\left\{\frac{1}{\kappa_{bhm}\lambda}, 1\right\} \varepsilon^{\frac{2}{\min(p-1,1)}} + c_1 \gamma^p C_7^p \varepsilon^{\frac{p}{\min(p-1,1)}}.$$

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

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

and the proof is completed. ■

The impact of imposing $\|g_k\| \leq \varepsilon_c$ to perform a series of criticality steps appears only when counting successful iterations. In fact, one would have instead $\|g_k\| \geq \min\{\varepsilon_c, \Delta_k/\lambda\}$ when $k \in \mathscr{S}$.

One possibility to fix the situation would be to select

$$\varepsilon_c \;\geq\; \mathscr{O}(\varepsilon^{\frac{1}{\min(p-1,1)}})$$

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 $\min\{\varepsilon_c, \Delta_k/\lambda\} = \Delta_k/\lambda$. 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 $\varepsilon_c\lambda$.

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

Theorem 4.1.3 *Under the conditions of Theorem 4.1.2, to achieve $\|\nabla f(x_{\bar{k}})\| \leq \varepsilon$, starting from k_0, Algorithm 4.1.1 takes at most $|\mathscr{N}(k_0,\bar{k})|$ other (not successful) iterations, where*

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

$$L_1 \;=\; -\log_\gamma(\gamma_{inc}), \quad L_2 \;=\; \log_\gamma\left(\frac{\gamma C_7 e}{\Delta_{k_0}}\right),$$

and C_7 is given in (4.6).

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

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

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

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

By Lemma 4.1.3, we have

$$|\mathscr{R}(k_0,\bar{k})| \;\leq\; L_1|\mathscr{S}(k_0,\bar{k})| + \log_\gamma\left(\frac{\gamma C_7}{\Delta_{k_0}}\right) - \frac{\log(\varepsilon^{-\frac{1}{\min(p-1,1)}})}{\log(\gamma)},$$

and thus, using $\log(x) \leq x-1$ for $x > 1$,

$$|\mathscr{R}(k_0,\bar{k})| \;\leq\; L_1|\mathscr{S}(k_0,\bar{k})| + L_2 - \log_\gamma(e)\varepsilon^{-\frac{1}{\min(p-1,1)}}. \tag{4.7}$$

It remains to count the iterations that are in $\mathscr{C}^{nr}$ and in $\mathscr{M}$. After an iteration in $\mathscr{C}^{nr}$ (the 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

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

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

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

Thus,

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

which combined with (4.7) 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

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

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 4.1.3, since there are no iterations in $\mathscr{R}$ until k_0, one has $k_0 \leq 4|\mathscr{S}(0,k_0-1)|$. To count the number of successful iterations up to k_0-1, we write, as in the proof of Theorem 4.1.2, 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 $\varepsilon < 1$, we obtain

$$k_0 \;\leq\; 4|\mathscr{S}(0,k_0-1)| \;\leq\; 4\frac{f(x_0)-f(x_{k_0})}{\min\{\Delta_0^2,\Delta_0^p\}L_0} \;\leq\; 4\frac{f(x_0)-f(x_{k_0})}{\min\{\Delta_0^2,\Delta_0^p\}L_0}\varepsilon^{-\frac{\max(p,2)}{\min(p-1,1)}}, \tag{4.8}$$

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 [23, Chapter 2] or the explanation about linear interpolation and regression models in Section 2.2).

Assumption 4.1.1 *The constants κ_{ef} and κ_{eg} in the definition of fully linear models satisfy $\kappa_{ef} = \mathscr{O}(\sqrt{n}L_{\nabla f})$ and $\kappa_{eg} = \mathscr{O}(\sqrt{n}L_{\nabla f})$, where n is the problem dimension and $L_{\nabla f}$ is the Lipschitz constant of the gradient of the objective function f.*

Theorems 4.1.2 and 4.1.3 and the bound on k_0 given above, together with Assumption 4.1.1, lead to the following result.

Theorem 4.1.4 *Let Assumptions 2.1.1, 2.1.3, 2.2.1, and 4.1.1 hold. To drive the norm of the gradient below $\varepsilon \in (0,1)$, Algorithm 4.1.1 takes at most*

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

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

Proof. It suffices to observe that for the constant L appearing in Theorem 4.1.2 we have

$$\frac{1}{L} \;=\; \mathscr{O}\left(C_7^{-\max(p,2)}\right) \;=\; \mathscr{O}\left((C_5+C_6)^{\frac{\max(p,2)}{\min(p-1,1)}}\right) = \mathscr{O}\left(\kappa^{\frac{\max(p,2)}{\min(p-1,1)}}\right),$$

with $\kappa = \max\{\kappa_{ef}, \kappa_{eg}\}$ and then to apply Assumption 4.1.1. ■

Algorithm 4.1.1 takes at most $\mathscr{O}(n)$ 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 4.1.1 *Let Assumptions 2.1.1, 2.1.3, 2.2.1, and 4.1.1 hold. To drive the norm of the gradient below $\varepsilon \in (0,1)$, Algorithm 4.1.1 takes at most*

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

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

4.2 Complexity in determining second-order stationary points

It is also possible to count the effort of derivative-free trust-region methods in the determination of second-order critical points. In fact, we will see in this section that most of what we did for the first-order case can be extended to the second-order case in a way that the analysis is carried out naturally.

The algorithm revisited

We start by reproducing in detail the second-order version of Algorithm 4.1.1. The changes are the expected ones at this point of the thesis, meaning the use of fully quadratic models, the use of the second-order model criticality measure (2.25), and an approximated solution of the trust-region subproblem satisfying a fraction of the eigenvalue decrease. For simplicity, we use here the notation

$$\tau_k \;=\; \lambda_{\min}(H_k).$$

We will make a simplification relatively to Algorithm 4.1.1, by not using the term $c_1\Delta_k^p$ in the definition of ρ_k — as we will not apply the second-order version in the context of non-smooth problems (a topic for future research).

Algorithm 4.2.1 Derivative-free trust-region method using fully quad. models (version WCC)

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 $0 \le \eta_0 \le \eta_1 < 1$ (with $\eta_1 \neq 0$), $\gamma \in (0,1)$, $\gamma_{inc} > 1$, and $\lambda > \beta > 0$. Set $k = 0$.

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

$$m_k(x_k) - m_k(x_k + s_k) \;\geq\; \frac{\kappa_{fod}}{2} \max\left\{ \|g_k\| \min\left\{ \frac{\|g_k\|}{\|H_k\|}, \Delta_k \right\}, -\tau_k \Delta_k^2 \right\} \tag{4.9}$$

(with $\kappa_{fod} \in (0,1]$), 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

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

If $\rho_k \geq \eta_1$ or if $\rho_k \geq \eta_0$ and m_k is fully quadratic, 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 quadratic on $B(x_k;\Delta_k)$,
- if such a certificate is not obtained, we say that m_k is not certifiably fully quadratic 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

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

Increment k by one and go to Step 1.

As in Algorithm 4.1.1, there are essentially five types of trust-region iterations resulting from Algorithm 4.2.1 (critical, successful, acceptable, unsuccessful, model-improvement), and the critical iterations are split in two types depending on whether the trust-region radius is reduced or not. The description of these iterations has been made after Algorithm 4.1.1. All the other considerations made there apply to Algorithm 4.2.1.

Global convergence

Let us recall the assumptions that we need for this section, the same as in Section 2.3 for the second-order case: Assumptions 2.1.1, 2.1.3, and 2.2.3. Again, given that substantial modifications in the

presentation of the algorithm are made relatively to the original description in [22] (Algorithm 2.3.3 in Section 2.3), it becomes necessary to redo the global convergence theory.

We will first show that the trust-region radius converges to zero, adapting the proof of Lemma 4.1.1 to the second-order case.

Lemma 4.2.1 *Let Assumptions 2.1.1 and 2.1.3 hold. Then*

$$\lim_{k\to+\infty} \Delta_k \;=\; 0.$$

Proof. First we assume that the number of successful iterations is finite. This part of the proof is verbatim equal to the one of Lemma 4.1.1 (except that "fully linear" there is now fully quadratic).

Let us assume now that $\mathscr{S}$ is infinite. When k is in $\mathscr{S}$, from (4.9) and Assumption 2.1.3, we have that

$$f(x_k) - f(x_{k+1}) \;\geq\; \eta_1 \frac{\kappa_{fod}}{2} \max\left\{ \|g_k\| \min\left\{ \frac{\|g_k\|}{\kappa_{bhm}}, \Delta_k \right\}, -\tau_k \Delta_k^2 \right\}.$$

Since the iteration is not critical, $\sigma_k^m \geq \Delta_k/\lambda$. When $\sigma_k^m = \|g_k\|$,

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta_1 \kappa_{fod}}{2\lambda} \Delta_k \min\left\{ \frac{\Delta_k}{\kappa_{bhm}\lambda}, \Delta_k \right\}, \tag{4.10}$$

and when $\sigma_k^m = -\tau_k$,

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta_1 \kappa_{fod}}{2\lambda} \Delta_k^3. \tag{4.11}$$

The proof can be completed exactly as in the proof of [22, Lemma 7.7] (see also [23, Lemma 10.20]): There are two subsequences of successful iterations, possibly overlapping, $\{k_i^1\}$, for which (4.10) holds, and $\{k_i^2\}$, for which (4.11) holds. Note that the union of these subsequences contains all successful iterations, and because $\mathscr{S}$ is infinite and f is bounded from below, either the corresponding subsequence $\{k_i^1\}$ (resp. $\{k_i^2\}$) is finite or the right hand side of (4.10) (resp. (4.11)) has to converge to zero. The conclusion is then that $\lim_{k\in\mathscr{S}} \Delta_k = 0$, and the proof is completed if all iterations are successful. Now, if there exists an iteration k that is not successful then, due to the way in which Δ_k is updated at Step 5 of Algorithm 4.2.1, we have $\Delta_k \leq \gamma_{inc}\Delta_{s_k}$, where s_k is the last successful iteration before k. Since $\lim_{s_k\in\mathscr{S}} \Delta_{s_k} = 0$, the proof is completed in this case as well. ■

The model criticality measure

$$\sigma^m(x) \;=\; \max\{\|\nabla m(x)\|, -\lambda_{\min}(\nabla^2 m(x))\} \tag{4.12}$$

provides an accurate approximation to the criticality measure $\sigma(x)$ defined in (2.8). The result is taken from [22, Lemma 7.2] (see also [23, Lemma 10.15]).

Lemma 4.2.2 *Let Δ be bounded by $\Delta_{\max}$. Suppose that Assumption 2.2.3 holds and m is a fully quadratic model on $B(x;\Delta)$. Then, we have that*

$$|\sigma(x) - \sigma^m(x)| \;\leq\; \kappa_\sigma \Delta, \tag{4.13}$$

where $\kappa_\sigma = \max\{\kappa_{eg}\Delta_{max}, \kappa_{eh}\}$.

Using this notation, note that σ_k^m in (2.25) coincides with $\sigma^m(x_k)$. The next step in the analysis is to prove that if the trust-region radius passes below a constant times this measure, then it is not further reduced:

Lemma 4.2.3 *Let Assumptions 2.1.3 and 2.2.3 hold. If k is an iteration for which Δ_k is reduced, then*

$$\sigma(x_k) \leq C_8 \Delta_k,$$

where

$$C_8 = \kappa_\sigma + C_{0,2} \quad \text{and} \quad C_{0,2} = \frac{1}{\min\left\{\beta, \frac{1}{\kappa_{bhm}}, \frac{\kappa_{fod}(1-\eta_1)}{4\kappa_{ef}\Delta_{\max}}, \frac{\kappa_{fod}(1-\eta_1)}{4\kappa_{ef}}\right\}}. \tag{4.14}$$

Proof. By assumption we have that $k \in \mathscr{R} = \mathscr{C}^r \cup \mathscr{A} \cup \mathscr{U}$. Let us suppose that $k \in \mathscr{A} \cup \mathscr{U}$. We will show first that

$$\sigma_k^m \leq C_{0,2} \Delta_k. \tag{4.15}$$

Assume by contradiction that (4.15) is false. From the definition (2.25) of σ_k^m, we have either $\sigma_k^m = \|g_k\|$ or $\sigma_k^m = -\lambda_{\min}(H_k) = -\tau_k$.

Suppose that $\sigma_k^m = \|g_k\|$. Given that $C_{0,2} \geq \frac{4\kappa_{ef}\Delta_{\max}}{\kappa_{fod}(1-\eta_1)}$, we then obtain

$$\sigma_k^m > \frac{4\kappa_{ef}\Delta_{\max}}{\kappa_{fod}(1-\eta_1)} \Delta_k.$$

From (4.9), Assumption 2.1.3, $C_{0,2} \geq \kappa_{bhm}$, we have

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{\kappa_{fod}}{2} \|g_k\| \Delta_k = \frac{\kappa_{fod}}{2} \sigma_k^m \Delta_k.$$

Hence, using (2.23) for both $s = 0$ and $s = s_k$,

$$\begin{aligned} 1 - \eta_1 &> \frac{4\kappa_{ef}\Delta_{\max}\Delta_k}{\kappa_{fod}\sigma_k^m} \\ &\geq \left| \frac{f(x_k + s_k) - m_k(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \right| + \left| \frac{f(x_k) - m_k(x_k)}{m_k(x_k) - m_k(x_k + s_k)} \right| \\ &\geq |\rho_k - 1|. \end{aligned}$$

Consider now the second case where $\sigma_k^m = -\tau_k$. From (4.9),

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{\kappa_{fod}}{2}(-\tau_k)\Delta_k^2 = \frac{\kappa_{fod}}{2} \sigma_k^m \Delta_k^2.$$

Hence, using similar arguments, given that $C_{0,2} \geq \frac{4\kappa_{ef}}{\kappa_{fod}(1-\eta_1)}$, and assuming that (4.15) is false, we then obtain

$$\begin{aligned} 1 - \eta_1 &> \frac{4\kappa_{ef}\Delta_k}{\kappa_{fod}\sigma_k^m} \\ &\geq |\rho_k - 1|. \end{aligned}$$

(Note that after having applied (2.23), Δ_k^2 canceled and there was no need to use the bound $\Delta_k \leq \Delta_{\max}$.)

In any of the cases, we concluded that $\rho_k > \eta_1$, implying that the iteration is successful and contradicting the fact that $k \in \mathscr{A} \cup \mathscr{U}$. We have thus proved (4.15) for $k \in \mathscr{A} \cup \mathscr{U}$.

To establish the result of the lemma when $k \in \mathscr{A} \cup \mathscr{U}$, it remains to use Lemma 4.2.2 and write

$$\sigma(x_k) \;\leq\; \sigma(x_k) - \sigma^m(x_k) + \sigma^m(x_k) \;\leq\; \kappa_\sigma \Delta_k + C_{0,2}\Delta_k.$$

Finally, suppose that $k \in \mathscr{C}^r$. Here the proof is exactly as in the proof of Lemma 4.1.2, with the use of (2.11) replaced by the use of (4.13). ■

As in the first-order case, a global convergence result follows directly from Lemma 4.2.3 and the asymptotic behavior of the trust-region radius.

Theorem 4.2.1 *Let Assumptions 2.1.1, 2.1.3, and 2.2.3 hold. Then*

$$\liminf_{k \to +\infty} \sigma(x_k) \;=\; 0.$$

Proof. By Lemma 4.2.1, there is an infinite subsequence of iterations where the trust-region radius is reduced, to which then we can apply Lemma 4.2.3. ■

Worst case complexity

In this section, we derive the worst-case complexity analysis of Algorithm 4.2.1. Given a tolerance $\varepsilon \in (0,1)$, we are interested in knowing how many iterations are needed to compute an iterate x_k such that $\sigma(x_k) \leq \varepsilon$. As in the first-order case, we start by stating the following auxiliary lemma whose proof is exactly the same as the proof of Lemma 4.1.3 (with $\nabla f(x_k)$ replaced by $\sigma(x_k)$).

Lemma 4.2.4 *Let Assumptions 2.1.3 and 2.2.3 hold. Let $\varepsilon \in (0,1)$. Let k_0 be the first iteration where Δ_k is reduced. For every iteration $k \geq k_0$ of the algorithm, if $\sigma(x_j) > \varepsilon$ for $j = k_0, \ldots, k$, then*

$$\Delta_k \;\geq\; \gamma C_9 \varepsilon,$$

where

$$C_9 \;=\; \min\left(1, C_8^{-1}\right), \tag{4.16}$$

with C_8 is given in (4.14).

The next step is to count the number of successful iterations.

Theorem 4.2.2 *Let Assumptions 2.1.1, 2.1.3, and 2.2.3 hold. Let k_0 be the index of the first iteration where Δ_k is reduced (which must exist from Lemma 4.2.1). Given any $\varepsilon \in (0,1)$, assume that $\sigma(x_{k_0}) > \varepsilon$ and let $\bar{k}$ be the first iteration after k_0 such that $\sigma(x_{\bar{k}}) \leq \varepsilon$. Then, to achieve $\sigma(x_{\bar{k}}) \leq \varepsilon$, starting from k_0, Algorithm 4.2.1 takes at most $|\mathscr{S}(k_0, \bar{k})|$ successful iterations, where*

$$|\mathscr{S}(k_0, \bar{k})| \;\leq\; \frac{f(x_{k_0}) - f_{low}}{\bar{L}} \varepsilon^{-3},$$

where

$$\bar{L} \;=\; \frac{\eta_1 \kappa_{fod} \gamma^2 C_9^2}{2\lambda} \min\left\{ \frac{1}{\kappa_{bhm}\lambda}, 1, \gamma C_9 \right\}$$

with C_9 given in (4.16).

Proof. For those iterations in $\mathscr{S}(k_0, \bar{k})$ we know that $\sigma_k^m \geq \Delta_k/\lambda$, thus either $\|g_k\| \geq \Delta_k/\lambda$ or $-\tau_k = -\lambda_{\min}(H_k) \geq \Delta_k/\lambda$. Using (4.9), one has

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

in the first case, and

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta_1 \kappa_{fod}}{2} \frac{\Delta_k}{\lambda} \Delta_k^2$$

in the second case. One now applies Lemma 4.2.4 and use the same logic as in the first-order case to obtain the desired result. ■

Now we count all iterations, after k_0, which are not successful (the proof is the same as the proof of Theorem 4.1.3).

Theorem 4.2.3 *Under the conditions of Theorem 4.2.2, to achieve $\sigma(x_{\bar{k}}) \leq \varepsilon$, starting from k_0, Algorithm 4.2.1 takes at most $|\mathscr{N}(k_0, \bar{k})|$ other (not successful) iterations, where*

$$|\mathscr{N}(k_0, \bar{k})| \;\leq\; (3 + 4\bar{L}_1)|\mathscr{S}(k_0, \bar{k})| + 4\left(\bar{L}_2 - \log_\gamma(e)\varepsilon^{-1}\right),$$

where

$$\bar{L}_1 \;=\; L_1 \;=\; -\log_\gamma(\gamma_{inc}), \quad \bar{L}_2 \;=\; \log_\gamma\left(\frac{\gamma C_9 e}{\Delta_{k_0}}\right),$$

and C_9 is given in (4.16).

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 σ below ε is $\mathscr{O}\left(\varepsilon^{-3}\right)$. It can also be easily shown that k_0 is also bounded by such a quantity. As in the first-order case, since there are no iterations in $\mathscr{R}$ until k_0, one has $k_0 \leq 4|\mathscr{S}(0, k_0 - 1)|$. Summing up all successful iterations up to k_0, and considering $\Delta_k \geq \Delta_0$ and $\varepsilon < 1$, we obtain (as in the proof of Theorem 4.2.2)

$$k_0 \;\leq\; 4|\mathscr{S}(0, k_0 - 1)| \;\leq\; 4\frac{f(x_0) - f(x_{k_0})}{\Delta_0^2 \bar{L}_0} \;\leq\; 4\frac{f(x_0) - f(x_{k_0})}{\Delta_0^2 \bar{L}_0}\varepsilon^{-3} \tag{4.17}$$

with

$$\bar{L}_0 \;=\; \frac{\eta_1 \kappa_{fod}}{2\lambda} \min\left\{ \frac{1}{\kappa_{bhm}\lambda}, 1, \Delta_0 \right\}.$$

Note that in the derivative-based trust-region setting [13] it is possible to prove that at most $\mathscr{O}(\max\{\varepsilon_g^{-2}\varepsilon_H^{-1}, \varepsilon_H^{-3}\})$ iterations are needed to determine a point x_k such that

$$\|\nabla f(x_k)\| \;\leq\; \varepsilon_g \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(x_k)) \;\geq\; -\varepsilon_H.$$

It is not clear, however, if such a result is still true in the derivative-free case.

As in the first-order case, 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. The constants κ_{ef}, κ_{eg}, and κ_{eh} in the definition of fully quadratic models depend on $\sqrt{p}$ and on the inverse of the norm of scaled versions of the interpolation matrix (see [22], [23, Chapter 3], or the explanation about quadratic interpolation and regression models in Section 2.2). Having in mind that $p \geq (n+1)(n+2)/2-1$ and ignoring the latter effect, one can suppose that κ_{ef}, κ_{eg}, and κ_{eh} are all $\mathscr{O}(n)$. Theorems 4.2.2 and 4.2.3 and the bound on k_0 given by (4.8), together with this assumption, lead to the following result.

Theorem 4.2.4 *Let Assumptions 2.1.1, 2.1.3, and 2.2.3 hold. Let us assume that κ_{ef}, κ_{eg}, and κ_{eh} are all $\mathscr{O}(n)$. To drive the value of σ below $\varepsilon \in (0,1)$, Algorithm 4.2.1 takes at most*

$$\mathscr{O}\left(n^3\varepsilon^{-3}\right)$$

iterations.

Proof. It suffices to observe that for the constant $\bar{L}$ appearing in Theorem 4.2.2 we have

$$\frac{1}{\bar{L}} = \mathscr{O}\left(C_9^{-3}\right) = \mathscr{O}\left(C_8^3\right) = \mathscr{O}\left(\kappa^3\right),$$

with $\kappa = \max\{\kappa_{ef}, \kappa_\sigma\}$ and one knows from Lemma 4.2.2 that $\kappa_\sigma = \max\{\kappa_{eg}\Delta_{max}, \kappa_{eh}\}$. ■

Algorithm 4.2.1 takes at most $\mathscr{O}(n^2)$ 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 4.2.1 *Let Assumptions 2.1.1, 2.1.3, and 2.2.3 hold. To drive the value of σ below $\varepsilon \in (0,1)$, Algorithm 4.2.1 takes at most*

$$\mathscr{O}\left(n^5\varepsilon^{-3}\right)$$

function evaluations.

Chapter 5

Derivative-free trust-region methods for non-smooth functions

Trust-region methods have been well studied for non-smooth functions by considering non-smooth trust-region subproblems which use the non-smooth structure of the function in a way that function and model share the same (generalized) derivatives (see [27] and [18, Chapter 11]).

The existing approaches for non-smooth DFO are essentially of one of the three following types: approximation by a family of smoothing functions (see [34, 45, 53]), where in practice the structure of non-smoothness must be known; direct use of directions asymptotically dense in the unit sphere (see [2, 4, 65]), where no structure is needed even for the theory; explicit use of a known type of smoothness, both in theory and in practice (see [7, 44] for minmax-type structure and [37] for composition of a smooth function with a non-smooth convex one, the latter approach using trust regions).

In this chapter we start by reviewing basic properties of non-smooth functions (Section 5.1) and of how to approximate them by a family of smooth functions (Section 5.2). Then we introduce two new approaches for optimizing non-smooth functions using trust-region methods without derivatives. The first one (Section 5.3) makes use of smoothing techniques, by applying trust-region methods to a sequence of smooth functions converging to the original one. A second approach (Section 5.4) is developed specifically for composite functions, where the non-smooth component of the function is moved to the trust-region subproblem. A numerical illustration of the relative performance of these methodologies is given in Section 5.5.

5.1 A review of basic concepts in non-smooth analysis

In this section we are going to review some basic results from non-smooth analysis that we need to better understand the rest of the chapter. The presentation follows Clarke [17], focusing only on $\mathbb{R}^n$.

A function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz near x if there exist scalars $L, \varepsilon > 0$ such that

$$|f(y) - f(z)| \leq L \| y - z \|$$

for all y, z in $B(x;\varepsilon)$. When f is Lipschitz near x, one can define the Clarke generalized directional derivative of f at x along the direction v by

$$f^\circ(x;v) \;=\; \limsup_{y\to x, t\downarrow 0} \frac{f(y+tv)-f(y)}{t}.$$

This definition is meant in the sense:

$$f^\circ(x;v) \;=\; \lim_{\varepsilon\to 0} \sup_{y\in B(x;\varepsilon), t\in B(0;\varepsilon)} \frac{f(y+tv)-f(y)}{t}.$$

If the function f is Lipschitz continuous near x with constant L, then $f^\circ(x;\cdot)$ is also Lipschitz continuous in $\mathbb{R}^n$ with the same constant L.

Suppose again that f is Lipschitz near $x \in \mathbb{R}^n$. The Clarke generalized subdifferential of f at x can then be defined by

$$\partial f(x) \;=\; \{s \in \mathbb{R}^n :\; f^\circ(x;v) \geq v^\top s,\; \forall v \in \mathbb{R}^n\}.$$

Moreover, it can be proved that

$$f^\circ(x;v) \;=\; \max\{v^\top s : s \in \partial f(x)\}. \tag{5.1}$$

One can define stationarity or a first-order necessary condition for general Lipschitz continuous functions using the Clarke generalized subdifferential or derivative. In fact, if f attains a local minimum at x_*, then $f^\circ(x_*;v) \geq 0$ for all $v \in \mathbb{R}^n$, or, equivalently, $0 \in \partial f(x_*)$.

When f is locally Lipschitz continuous, the Clarke generalized subdifferential is a nonempty convex set and, as a set-valued mapping, is closed and locally bounded. A mean-value theorem can also be formulated for locally Lipschitz functions using the Clarke generalized subdifferential, a result known by the Lebourg mean-value theorem. In fact, if x and y are points in $\mathbb{R}^n$ and if f is Lipschitz continuous in an open set containing the line segment $[x,y]$, then there exists a point z in (x,y) such that

$$f(y)-f(x) \;=\; s(z)^\top (y-x),$$

for some $s(z) \in \partial f(z)$. When f is convex, $\partial f(x)$ coincides with the subdifferential of convex analysis, i.e., the set of vectors $s \in \mathbb{R}^n$ satisfying

$$f(x+u)-f(x) \;\geq\; u^\top s,\;\; \forall u \in \mathbb{R}^n.$$

There are intermediate degrees from non-smoothness to smoothness, from being Lipschitz continuous (with a Clarke generalized directional derivative) to being continuous differentiable. For instance, a function f is regular at x if it has directional derivatives $f'(x;v)$ for all v and they coincide with $f^\circ(x;v)$. It can be proved that, if a convex function is Lipschitz continuous near a point then it is regular at that point (see [17, Proposition 2.3.6]). A function is said strictly differentiable at x (when Lipschitz continuous) if there exists a vector ζ (called $\nabla f(x)$) such that $f^\circ(x;v) = \nabla f(x)^\top v$,

for all v, in which case $\partial f(x) = \{\nabla f(x)\}$. Such a strict differentiability implies regularity. Continuous differentiability implies, in turn, strict differentiability.

The Clarke generalized subdifferential admits a nice geometrical characterization as the polar of a tangent cone to the epigraph of the function. For this purpose, let us make some definitions first. A vector d in $\mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point x in the closure of Ω if for every sequence $\{y_k\}$ of elements of Ω that converges to x and for every sequence of positive real numbers $\{t_k\}$ converging to zero, there exists a sequence of vectors $\{w_k\}$ converging to d such that $y_k + t_k w_k \in \Omega$. We denote such a set of tangent vectors by $T_\Omega(x)$. We define the normal cone to Ω at x by polarity with $T_\Omega(x)$:

$$N_\Omega(x) \;=\; \{\zeta \in \mathbb{R}^n : \zeta^\top v \leq 0, \;\; \forall v \in T_\Omega(x)\}.$$

In turn, the epigraph of a real-valued function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is the following subset of $\mathbb{R}^n \times \mathbb{R}$:

$$\text{epi}(f) \;=\; \{(x,r) \in \mathbb{R}^n \times \mathbb{R} : f(x) \leq r\}.$$

It is then known that an element ζ of $\mathbb{R}^n$ belongs to $\partial f(x)$ if and only if $(\zeta, -1)$ belongs to $N_{\text{epi}(f)}(x, f(x))$.

We recall also here Rademacher's Theorem that states that a Lipschitz continuous function on an open set of $\mathbb{R}^n$ is differentiable almost everywhere, in the sense of the Lebesgue measure. Let us denote by Ω_f the set of points at which a given function f fails to be differentiable. Suppose that f is Lipschitz continuous near x, and let S be any set of Lebesgue measure 0 in $\mathbb{R}^n$. Then, the Clarke generalized subdifferential also admits the following characterization:

$$\partial f(x) \;=\; \text{co}\{\lim \nabla f(x_i) : x_i \to x, x_i \notin S, x_i \notin \Omega_f\},$$

where co denotes the convex hull operator.

Consider the simple example $f(x) = |x|$ for $x \in \mathbb{R}$. Let $x = 0$. By the geometrical characterization given before, relating $\partial f(0)$ to the normal cone of the epigraph of f at $(0, f(0))$, it follows directly that $\partial f(0) = \partial |\cdot|(0) = [-1, 1]$. Then, from (5.1) it results that $f^\circ(0; v) = |v|$.

5.2 Smoothing of non-smooth functions

Given a possibly non-smooth objective function f, it is of interest to us the definition of a smoothing function (see [15, 70]):

Definition 5.2.1 *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 the next 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, co $G_{\tilde{f}}(x_*) = \partial f(x_*)$, where $\partial f(x_*)$ denotes the Clarke subdifferential of f at x_*, a result known as gradient consistency and that follows from [62, Theorem 9.67]). It is known that $\partial f(x_*)$ is generally contained in the convex hull of $G_{\tilde{f}}(x_*)$. For certain smoothing functions the other inclusion also occurs, i.e., $G_{\tilde{f}}(x_*)$ is included in $\partial f(x_*)$ (for more details see the summary in [34]). Thus, in those cases of gradient consistency, the smoothing trust-region methods are capable of generating a sequence of iterates converging to Clarke stationary points.

We have especially in mind the minimization of 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. An example that finds many applications is when h is given by the ℓ_1-norm, $h(\cdot) = \|\cdot\|_1$. In the rest of this section we will explain how to build a smoothing function for $f = h(F) = \|F\|_1$ with all the desired properties, including gradient consistency.

We start by describing how to build a smoothing function for the absolute value, using the approach suggested by Chen and Zhou [16]:

$$\tilde{s}(t,\mu) \;=\; \int_{-\infty}^{+\infty} |t - \mu\tau| \rho(\tau) d\tau, \tag{5.2}$$

where $\rho : \mathbb{R}^n \to [0,+\infty)$ is a piecewise continuous density function with a finite number of pieces satisfying

$$\rho(\tau) \;=\; \rho(-\tau) \quad \text{and} \quad \int_{-\infty}^{+\infty} |\tau| \rho(\tau) d\tau \;<\; +\infty.$$

Let $\kappa = \int_{-\infty}^{+\infty} |\tau| \rho(\tau) d\tau$. The following proposition, which is a special case of [16, Proposition 3.1], describes the relevant properties of $\tilde{s}(t,\mu)$.

Proposition 5.2.1 *The function $\tilde{s}(t,\mu)$ defined by (5.2) has the following properties:*
(i) $\tilde{s}(t,\mu) = \tilde{s}(-t,\mu)$ for $t \in \mathbb{R}$, that is, $\tilde{s}(\cdot,\mu)$ is symmetric.
(ii) $\tilde{s}(\cdot,\mu)$ is continuously differentiable on $\mathbb{R}$, and its derivative can be given by

$$\tilde{s}'(t,\mu) \;=\; 2\int_0^{\frac{t}{\mu}} \rho(\tau) d\tau.$$

(iii) $\tilde{s}(\cdot,\mu)$ converges uniformly to $|t|$ on $\mathbb{R}$ with

$$|\tilde{s}(t,\mu) - |t|| \;\leq\; \kappa\mu.$$

(iv) The set of limits of the derivatives $\tilde{s}'(t,\mu)$ coincides with the Clarke subdifferential of the absolute

value, that is,

$$\left\{ \lim_{t\to 0,\mu\downarrow 0} \tilde{s}'(t,\mu) \right\} = [-1,1] = \partial|\cdot|(0) \quad \textit{and} \quad \lim_{t\to t_*,\mu\downarrow 0} \tilde{s}'(t,\mu) = \begin{cases} 1 & t_* > 0, \\ -1 & t_* < 0. \end{cases}$$

Moreover, one has

$$\lim_{\mu\downarrow 0} \tilde{s}'(t,\mu) = \begin{cases} 1 & t > 0, \\ 0 & t = 0, \\ -1 & t < 0. \end{cases}$$

(v) For any fixed $\mu > 0$, $\tilde{s}'(t,\mu)$ is Lipschitz continuous with constant $2\kappa_0/\mu$, where κ_0 is an upper bound for ρ.

If one considers the following uniform density function [16],

$$\rho(\tau) = \begin{cases} 1 & \text{if } \tau \in [-\frac{1}{2}, \frac{1}{2}], \\ 0 & \text{otherwise,} \end{cases}$$

then, using (5.2), the smoothing function for $|\cdot|$ corresponding to this density function is

$$\tilde{s}(t,\mu) = \begin{cases} \frac{t^2}{\mu} + \frac{\mu}{4} & \text{if } t \in [-\frac{\mu}{2}, \frac{\mu}{2}], \\ |t| & \text{otherwise,} \end{cases}$$

with gradient given by

$$\tilde{s}'(t,\mu) = \begin{cases} \frac{2t}{\mu} & \text{if } t \in [-\frac{\mu}{2}, \frac{\mu}{2}], \\ \text{sign}(t) & \text{otherwise.} \end{cases}$$

The Lipschitz constant of $\tilde{s}'(\cdot,\mu)$ is $2/\mu$.

Based on the above given smoothing function for $|\cdot|$, the following smoothing function for $\|F(\cdot)\|_1$ has been introduced in [34]

$$\tilde{F}(x,\mu) = \sum_{i=1}^{m} \tilde{s}(F_i(x),\mu). \tag{5.3}$$

Using the properties of $\tilde{s}$ given in Proposition 5.2.1 and the use of non-smooth calculus rules for regular functions, it has been proved in [34] that $\tilde{F}$ is indeed a smoothing function for $\|F(\cdot)\|_1$, satisfying the gradient consistency property and exhibiting a Lipschitz continuous gradient with constant of the order of $1/\mu$.

Theorem 5.2.1 *Let $\tilde{F}(x,\mu) = \sum_{i=1}^{m} \tilde{s}(F_i(x),\mu)$ be defined as in (5.3). Then*
(i) $\tilde{F}$ is a smoothing function for $\|F\|_1$.
(ii) $\tilde{F}(\cdot,\mu)$ satisfies the gradient consistent property, that is,

$$\left\{ \lim_{x\to x_*,\mu\downarrow 0} \nabla\tilde{F}(x,\mu) \right\} = \partial\|F\|_1(x_*).$$

(iii) For each μ, $\nabla\tilde{F}(\cdot,\mu)$ is Lipschitz continuous with a Lipschitz constant of the order of $1/\mu$.

Theorem 5.2.1 tells us, in fact, that $\tilde{F}(\cdot,\mu)$ is a continuously differentiable smoothing function for $f(\cdot) = \|F(\cdot)\|_1$, satisfying $G_{\tilde{F}}(x_*) = \partial f(x_*)$ and for which the gradient is Lipschitz continuous with constant $\mathcal{O}(1/\mu)$.

Other examples of smoothing functions with the same desired properties are given using Gaussian densities [53].

5.3 Smoothing trust-region methods without derivatives

In this section, following what has been done in [34] 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 algorithm

The idea is simple and consists of the application of Algorithm 4.1.1 to the smoothing function for decreasing values of the smoothing parameter μ. Each outer or main iteration (Algorithm 4.1.1 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 5.3.1 (Smoothing trust-region method)

Initialization

Choose x_0 with $f(x_0) < +\infty$, $\Delta_0 > 0$, $\mu_0 > 0$, and $\sigma \in (0,1)$.

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

1. **Trust-region method for a fixed smoothing parameter:** Apply Algorithm 4.1.1 to $\tilde{f}(\cdot,\mu_k)$ (starting from $y_{0,k} = x_k$) 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 $x_{k+1} = y_{j_k,k}$ 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 5.3.1 will stop under a criterion of the form $\mu_k \leq \mu_{tol}$, where $\mu_{tol} \in (0, \mu_0)$.

Global convergence

We will analyze the global convergence of the smoothing trust-region method (Algorithm 5.3.1) under the following assumptions, which are the natural counterparts, for the smoothing function, of the ones assumed in the smooth cases of Chapters 2 and 4.

Assumption 5.3.1 *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 (2.10), with $L(y_{0,k}) = \{y \in \mathbb{R}^n : \tilde{f}(y,\mu_k) \leq \tilde{f}(y_{0,k},\mu_k)\}$.*

Assumption 5.3.2 *For all k: the functions $\tilde{f}(\cdot,\mu_k)$ are bounded below in $L(y_{0,k})$.*

Each inner iteration of Algorithm 5.3.1 consists of one iteration of Algorithm 4.1.1 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 Chapters 2 and 4, we will require all these model Hessians to be uniformly bounded.

Assumption 5.3.3 *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 5.3.1 *Let Assumptions 5.3.2 and 5.3.3 hold. Then the smoothing parameter goes to zero:*

$$\lim_{k\to\infty}\mu_k \;=\; 0.$$

Proof. For each k, one knows, from Lemma 4.1.1, 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 5.3.2 *Let Assumptions 5.3.2 and 5.3.3 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 5.3.1 and the fact that $r(\mu_k)>\Delta_{j_k+1,k}=\gamma\Delta_{j_k,k}$. ■

Global convergence of Algorithm 5.3.1 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 $L_{\nabla\tilde{f}}(\mu)=\mathcal{O}(1/\mu)$. As we have seen in Section 5.2, there are smoothing functions satisfying this property as well as gradient consistency, such as the smoothing function for composite functions of the type $\|F\|_1$ where F is smooth.

Theorem 5.3.3 *Consider the application of Algorithm 5.3.1 and suppose that $\tilde{f}$ is a smoothing function for f. Let Assumptions 5.3.1, 5.3.2, and 5.3.3 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*

$$\lim_{k\to+\infty}\|\nabla\tilde{f}(x_k,\mu_k)\| \;=\; 0 \tag{5.4}$$

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

Proof. For each k, $x_{k+1} = y_{j_k,k}$, where j_k is an iteration such that the trust-region radius is reduced. Thus, in view of Lemma 4.1.2, we have

$$\|\nabla \tilde{f}(x_k,\mu_k)\| \;\leq\; C_5(\mu_k)\Delta_{j_k,k} + C_6\Delta_{j_k,k}^{p-1},$$

where now $C_5 = C_5(\mu_k)$ depends on μ_k through the dependence of $L_{\nabla \tilde{f}}(\mu_k)$. Since $C_5(\mu_k) = \mathscr{O}(\tilde{\kappa}_{eg}) = \mathscr{O}(L_{\nabla \tilde{f}}(\mu_k))$, where $\tilde{\kappa}_{eg}$ is the constant in the error bound (2.12) for the gradient of the model of the smoothing function $\tilde{f}$, and $r(\mu_k) > \Delta_{j_k+1,k} = \gamma\Delta_{j_k,k}$, one obtains

$$\|\nabla \tilde{f}(x_k,\mu_k)\| \;\leq\; \mathscr{O}(L_{\nabla \tilde{f}}(\mu_k))r(\mu_k) + C_6\Delta_{j_k,k}^{p-1}.$$

Then, due to Theorems 5.3.1 and 5.3.2, we obtain (5.4) and the proof is completed. ■

If one considers a smoothing function $\tilde{f}$ for which $L_{\nabla \tilde{f}}(\mu) = \mathscr{O}(1/\mu)$, it suffices to choose $r(\mu) = \mu^q$, with $q > 1$, to successfully apply Theorem 5.3.3.

As a consequence of the above result, when the smoothing function of f satisfies the gradient consistent property at a limit point x_*, yielding $G_{\tilde{f}}(x_*) \subseteq \partial f(x_*)$, x_* is a Clarke stationary point of the function f.

Worst case complexity

We also follow here the same steps as in [34] and start by first counting the number of inner iterations of Algorithm 5.3.1 to drive the smoothing parameter below a given threshold.

Theorem 5.3.4 *Consider the application of Algorithm 5.3.1 using the term $c_1\Delta^p$ when calling Algorithm 4.1.1 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 5.3.1, 5.3.2, and 5.3.3 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 5.3.1 takes at most $\mathscr{O}(|\log(\xi)|\xi^{-pq})$ 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 5.3.1 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 order of one (see (4.8)).

One has, for a successful iteration (j,k), that

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

Since $\Delta_{j,k} \geq c_2\mu_k^q$,

$$\tilde{f}(y_{j,k},\mu_k) - \tilde{f}(y_{j+1,k},\mu_k) \;\geq\; c_1c_2^p\mu_k^{pq}.$$

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

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

Similar to the first part of the proof of Theorem 4.1.3, the number of the other inner iterations is bounded as follows (remember that $0 < \gamma < 1$)

$$|\mathscr{R}_k(j_{0,k}, j_k)| \;\leq\; (3+4L_1)|\mathscr{S}_k(j_{0,k}, j_k)| - \log_\gamma(\Delta_{j_{0,k},k}) + \log_\gamma(\Delta_{j_k,k}).$$

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 $p > 1$, $\log_\gamma(\Delta_{j_k,k}) = \mathscr{O}(\xi^{-pq})$. We conclude that the maximum number of iterations needed in each inner loop minimization is $\mathscr{O}(\xi^{-pq})$.

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

$$\bar{k} \;\geq\; \frac{\log(\xi) - \log(\mu_0)}{\log(\sigma)},$$

and the proof is completed. ■

As we have seen in Section 5.2, there are situations where the Lipschitz constant of the gradient of the smoothing function is of the order of $1/\mu$. 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 5.3.5 *Let all assumptions of Theorem 5.3.4 hold and assume also that $L_{\nabla\tilde{f}}(\mu_k) = \mathscr{O}(1/\mu_k)$. 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 4.1.1.*

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

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

Proof. From Lemma 4.1.2 and $\Delta_{j_k,k} = \Delta_{j_k+1,k}/\gamma < (c_2/\gamma)\mu_k^q$, one has

$$\begin{aligned}\|\nabla\tilde{f}(x_{\bar{k}}, \mu_{\bar{k}})\| \;&\leq\; C_5\Delta_{j_{\bar{k}}} + C_6\Delta_{j_{\bar{k}}}^{p-1} \\ &\leq\; C_5(c_2/\gamma)\mu_{\bar{k}}^q + C_6(c_2/\gamma)^{p-1}\mu_{\bar{k}}^{(p-1)q}.\end{aligned}$$

The proof is completed by noting that $C_5 = \mathscr{O}(\tilde{\kappa}) = \mathscr{O}(\sqrt{n}L_{\nabla\tilde{f}}) = \mathscr{O}(\sqrt{n}/\mu)$ 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 $\mathscr{O}(\sqrt{n}\xi)$. We are thus finally ready to state a worst-case complexity bound for driving both the norm of the smoothing gradient and the smoothing parameter below a common threshold.

Corollary 5.3.1 *Under the assumptions of Theorem 5.3.5 and when $q = 2$ and $p = \frac{3}{2}$, Algorithm 5.3.1 takes at most $\mathscr{O}\left(|\log(\xi)|\xi^{-3}\right)$ iterations (and at most $\mathscr{O}\left(n|\log(\xi)|\xi^{-3}\right)$ function evaluations) to*

reduce the smoothing parameter below $\xi \in (0,1)$*, ending such process with*

$$\|\nabla \tilde{f}(x_{\bar{k}}, \mu_{\bar{k}})\| = \mathcal{O}(\sqrt{n}\xi). \tag{5.5}$$

Equivalently, the number of iterations needed to reach $\|\nabla \tilde{f}(x_{\bar{k}}, \mu_{\bar{k}})\| \leq \varepsilon$ and $\mu_{\bar{k}} \leq \xi = \varepsilon/(\sqrt{n}C)$, where $C > 0$ is the constant that multiplies $\sqrt{n}\xi$ in the right hand side of (5.5), is

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

leading to the following overall worst case complexity bound in terms of the number of function evaluations

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

5.4 Derivative-free 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 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.

Fully linear models revisited

Let $x_0 \in \mathbb{R}^n$ be a starting point for the trust-region methods considered in this section. 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 $L(x_0)$ of the form (2.5). As we have seen several times in this thesis, at each iteration of such methods, the function is sampled at trial points that may fall outside of the level set $L(x_0)$, and thus the set in which the function is sampled is taken as $L_{enl}(x_0)$, see (2.10). It is in $L_{enl}(x_0)$ that F is assumed smooth:

Assumption 5.4.1 *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), the models of F need to be assumed as accurate as first-order Taylor models, in the sense of being fully linear. It is further assumed that such models can be made fully linear in a finite number of model-improvement steps. We adapt below the Definition 2.2.1 of fully linear models for the case of vectorial functions, where ℓ can be greater than 1.

Definition 5.4.1 *Let a function* $F = (f_1, \ldots, f_\ell) : \mathbb{R}^n \to \mathbb{R}^\ell$*, that satisfies Assumption 5.4.1, 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*

   $$\max_{1 \le i \le \ell} \|\nabla f_i(x+s) - \nabla m_i(x+s)\| \;\le\; \kappa_{eg}\,\Delta, \quad \forall s \in B(0;\Delta), \tag{5.6}$$

   and

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

   $$\max_{1 \le i \le \ell} |f_i(x+s) - m_i(x+s)| \;\le\; \kappa_{ef}\,\Delta^2, \quad \forall s \in B(0;\Delta). \tag{5.7}$$

   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 5.4.1 coincides with Definition 2.2.1.

The algorithm

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 $\min_{\|s\| \le \Delta} l(x,s)$, 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

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

$\Psi(x,1)$ was used in [12] 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 [67, Lemma 2.1].

In this thesis, 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 $m(x+s)$ is convex and fully linear in the sense of Definition 5.4.1. One possibility to compute such a model is to set $m(x+s) = F(x) + J^m(x)s$, where the lines of the matrix $J^m(x)$ are the transposes of the simplex gradients of the components of F at x (see [23, Chapter 2] or the explanation in Section 2.2). 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 trust-region radius. This result was proved originally in [37, 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 (5.7),

$$\begin{aligned}
\Psi(x+t,1) - \Psi^m(x+t,1) \;&\leq\; l(x+t,0) - l(x+t,s_t) - [l^m(x+t,0) - l^m(x+t,s_t)] \\
&\leq\; h(F(x+t)) - h(m(x+t)) + h(m(x+t+s_t)) - h(F(x+t+s_t)) \\
&\leq\; (2L_h\kappa_{ef}\Delta_{max})\,\Delta.
\end{aligned}$$

In the case $\Psi(x+t,1) \leq \Psi^m(x+t,1)$, it can be proved similarly that $\Psi^m(x+t,1) - \Psi(x+t,1) \leq (2L_h\kappa_{ef}\Delta_{max})\Delta$. Therefore, we have

$$|\Psi(x+t,1) - \Psi^m(x+t,1)| \;\leq\; \kappa_\Psi\Delta, \quad \forall t \in B(0;\Delta), \quad \text{with} \quad \kappa_\Psi \;=\; 2L_h\kappa_{ef}\Delta_{max}. \tag{5.8}$$

A derivative-free trust region algorithm for composite functions can be stated in the same vein as it was done in Algorithm 4.1.1 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 4.1.1). There is no need now to consider the term $c_1\Delta_k^p$ in ρ_k, as its inclusion in Algorithm 4.1.1 was primarily done for deriving the complexity bounds for the smoothing trust-region approach of Section 5.3.

Algorithm 5.4.1 Derivative-free trust-region method (for composite functions)

Initialization: Same as in Algorithm 4.1.1 but setting $c_1 = 0$.

Step 1 (criticality step): Same as in Algorithm 4.1.1 but with g_k replaced by $\Psi_k^m = \Psi^m(x_k,1)$.

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 4.1.1 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 4.1.1.

Step 5 (trust-region radius update): Same as in Algorithm 4.1.1.

Similar to Algorithm 4.1.1, there are six types of iterations and we will use the same notation as in Section 4.1. 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.

Global convergence

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

Assumption 5.4.2 *The function $h:\mathbb{R}^\ell \to \mathbb{R}$ is convex, globally Lipschitz continuous, with Lipschitz constant $L_h > 0$, and bounded from below (by f_{low}).*

The following lemma and its proof are an adaptation of Lemma 2.1 in [12].

Lemma 5.4.1 *Let Assumption 5.4.2 hold. Then*

$$\Psi^m(x_k,\Delta_k) \;\geq\; \min\{\Delta_k,1\}\Psi_k^m.$$

Proof. When $\Delta_k > 1$, from $\min\limits_{\|s\|\leq 1} l^m(x_k,s) \geq \min\limits_{\|s\|\leq\Delta} l^m(x_k,s)$, 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 worst-case complexity 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 5.4.2 *Let Assumption 5.4.2 hold. Then*

$$\lim_{k\to+\infty} \Delta_k \;=\; 0.$$

Proof. The only differences from the proof of Lemma 4.1.1 lie in the use of the predicted decrease. Now, when $k \in \mathscr{S}$, we have

$$f(x_k) - f(x_{k+1}) \;\geq\; \eta_1\Psi^m(x_k,\Delta_k),$$

then by using Lemma 5.4.1,

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

and due to $\Psi_k^m \geq \Delta_k/\lambda$ (since the iteration is not in $\mathscr{C}$),

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

■

In the following lemma, which can be seen as a combination of Lemma 4.1.2 and Lemma 2.2 in [12], we bound the criticality measure by a constant multiple of the trust-region radius.

Lemma 5.4.3 *Let Assumptions 5.4.1 and 5.4.2 hold. If k is an iteration for which Δ_k is reduced, then*

$$\Delta_k \;\geq\; \min\left\{ \frac{1}{\kappa_\Psi \lambda + 1} \min\{\sqrt{C_{10}\Psi_k}, C_{10}\Psi_k\}, \frac{1}{\kappa_\Psi + 1/\beta}\Psi_k \right\},$$

where

$$C_{10} \;=\; \frac{1-\eta_1}{2L_h\kappa_{ef}}, \tag{5.9}$$

and κ_Ψ comes from (5.8).

Proof. By assumption we have that $k \in \mathscr{R} = \mathscr{C}^r \cup \mathscr{A} \cup \mathscr{U}$. Let us suppose that $k \in \mathscr{A} \cup \mathscr{U}$. To later arrive at a contradiction, suppose that

$$\Delta_k \;<\; \min\{\sqrt{C_{10}\Psi_k^m}, C_{10}\Psi_k^m\}. \tag{5.10}$$

Using (2.12) and Lemma 5.4.1, 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 $\Delta_k \leq C_{10}\Psi_k^m$,

$$|\rho_k - 1| \;\leq\; \frac{(2L_h\kappa_{ef})\Delta_k}{\Psi_k^m} \;\leq\; \frac{(2L_h\kappa_{ef})C_{10}\Psi_k^m}{\Psi_k^m} \;=\; 1-\eta_1.$$

If $\Delta_k > 1$, then, from $\Delta_k \leq \sqrt{C_{10}\Psi_k^m}$

$$|\rho_k - 1| \;\leq\; \frac{(2L_h\kappa_{ef})\Delta_k^2}{\Psi_k^m} \;\leq\; \frac{(2L_h\kappa_{ef})C_{10}\Psi_k^m}{\Psi_k^m} \;=\; 1-\eta_1.$$

We then obtain $\rho_k \geq \eta_1$ implying $k \in \mathscr{S}$, which contradicts $k \in \mathscr{A} \cup \mathscr{U}$. Thus, (5.10) is not true. Now, from (5.8) and the fact that k is not in $\mathscr{C}$,

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

and thus, $\Psi_k^m \geq \Psi_k/(\kappa_\Psi\lambda+1)$. Hence, since $\Delta_k \geq \min\{\sqrt{C_{10}\Psi_k^m}, C_{10}\Psi_k^m\}$, we have

$$\Delta_k \;\geq\; \frac{\min\{\sqrt{C_{10}\Psi_k}, C_{10}\Psi_k\}}{\kappa_\Psi\lambda+1}.$$

If $k \in \mathscr{C}^r$, then similarly to the last part of the proof of Lemma 4.1.2 (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 $\Delta_k \geq \Psi_k/(\kappa_\Psi + 1/\beta)$. ■

As in Theorem 4.2.1, a global convergence result can then be easily proved at this point of the analysis.

Theorem 5.4.1 *Let Assumptions 5.4.1 and 5.4.2 hold. Then*

$$\liminf_{k \to +\infty} \Psi_k \;=\; 0.$$

Proof. By Lemma 5.4.2, there is an infinite subsequence of iterations where the trust-region radius is reduced, to which then we can apply Lemma 5.4.3. ■

Worst case complexity

We proceed by stating the analog of Lemma 4.1.3.

Lemma 5.4.4 *Let Assumptions 5.4.1 and 5.4.2 hold. Let $\varepsilon \in (0,1)$. Let k_0 be the first iteration where Δ_k is reduced. For every iteration $k \geq k_0$ of the algorithm, if $\Psi_j > \varepsilon$ for $j = k_0, \ldots, k$, then*

$$\Delta_k \;\geq\; \gamma C_{11} \varepsilon.$$

where

$$C_{11} \;=\; \min\left\{ \frac{\min\{\sqrt{C_{10}}, C_{10}\}}{\kappa_\Psi \lambda + 1}, \frac{1}{\kappa_\Psi + 1/\beta} \right\} \tag{5.11}$$

and C_{10} is given in (5.9).

Proof. When $k \in \mathscr{R}$, it follows directly from Lemma 5.4.3, $\Psi_k > \varepsilon$, and $\varepsilon < 1$, that $\Delta_k \geq C_{11}\varepsilon$. When $k \notin \mathscr{R}$, the argument is the same as in the last paragraph of the proof of Lemma 4.1.3. ■

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

Theorem 5.4.2 *Let Assumptions 5.4.1 and 5.4.2 hold. Let k_0 be the index of the first iteration where Δ_k is reduced (which must exist from Lemma 5.4.2). Given any $\varepsilon \in (0,1)$, assume that $\Psi_{k_0} > \varepsilon$ and let $\bar{k}$ be the first iteration after k_0 such that $\Psi_{\bar{k}} \leq \varepsilon$. Then, to achieve $\Psi_{\bar{k}} \leq \varepsilon$, starting from k_0, Algorithm 5.4.1 takes at most $|\mathscr{S}(k_0, \bar{k})|$ successful iterations, where*

$$|\mathscr{S}(k_0, \bar{k})| \leq \frac{\lambda (f(x_{k_0}) - f_{low})}{\eta_1 \min\{\gamma C_{11}, 1\} \gamma C_{11}} \varepsilon^{-2},$$

where C_{11} given in (5.11).

Proof. Let $k \geq k_0$ be the index of a successful iteration. Using Lemma 5.4.1, $\Psi_k^m \geq \Delta_k/\lambda$, Lemma 5.4.4, and $\varepsilon \in (0,1)$, we obtain

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1 \Psi^m(x_k, \Delta_k) \\ &\geq \eta_1 \min\{\Delta_k, 1\} \Psi_k^m \\ &\geq \eta_1 \min\{\Delta_k, 1\} \frac{\Delta_k}{\lambda} \\ &\geq \frac{\eta_1}{\lambda} \min\{\gamma C_{11}\varepsilon, 1\} \gamma C_{11}\varepsilon \\ &\geq \frac{\eta_1}{\lambda} \min\{\gamma C_{11}, 1\} \gamma C_{11}\varepsilon^2. \end{aligned}$$

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

$$f(x_{k_0}) - f(x_{\bar{k}}) \;\geq\; |\mathscr{S}(k_0, \bar{k})| \frac{\eta_1}{\lambda} \min\{\gamma C_{11}, 1\} \gamma C_{11}\varepsilon^2,$$

and the proof is completed. ■

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

Theorem 5.4.3 *Let Assumptions 5.4.1 and 5.4.2 hold. Let k_0 be the index of the first iteration where Δ_k is reduced (which must exist from Lemma 5.4.2). Given any $\varepsilon \in (0,1)$, assume that $\Psi_{k_0} > \varepsilon$ and let $\bar{k}$ be the first iteration after k_0 such that $\Psi_{\bar{k}} \leq \varepsilon$. Then, to achieve $\Psi_{\bar{k}} \leq \varepsilon$, starting from k_0, Algorithm 5.4.1 takes at most $|\mathscr{N}(k_0, \bar{k})|$ other (not successful) iterations, where*

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

where C_{11} is given in (5.11),

$$L_3 = -\log_\gamma(\gamma_{inc}), \quad \textit{and} \quad L_4 = \log_\gamma\left(\frac{\gamma C_{11} e}{\Delta_{k_0}}\right).$$

Proof. The proof, except using Lemma 5.4.4 instead of Lemma 4.1.3, follows along the lines of that of Theorem 4.1.3. ■

The number of iterations necessary to achieve the first iteration k_0 (where the trust-region radius is reduced) is $\mathscr{O}(1)$, and thus k_0 is of the order of $\mathscr{O}\left(\varepsilon^{-2}\right)$, and the explanation is similar to the one for the smooth case discussed after Theorem 4.1.3. 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 dependance 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 5.4.3 *The constants κ_{ef} and κ_{eg} in the definition of fully linear models satisfy $\kappa_{ef} = \mathscr{O}(\sqrt{n}L_J)$ and $\kappa_{eg} = \mathscr{O}(\sqrt{n}L_J)$, 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 5.4.4 *Let Assumptions 5.4.1, 5.4.2, and 5.4.3 hold. To drive* Ψ *below* $\varepsilon \in (0,1)$*, Algorithm 5.4.1 takes at most* $\mathcal{O}\left(n\varepsilon^{-2}\right)$ *iterations.*

Proof. The proof is similar to that of Theorem 4.1.4. ■

The dependence of the bound on L_{J_F} was omitted but is $L_{J_F}^2$ as in Theorem 4.1.4 when $p = 2$.

Corollary 5.4.1 *Let Assumptions 5.4.1, 5.4.2, and 5.4.3 hold. To drive* Ψ *below* $\varepsilon \in (0,1)$*, Algorithm 5.4.1 takes at most* $\mathcal{O}\left(\ell n^2 \varepsilon^{-2}\right)$ *function evaluations.*

It can then be seen that, in terms of ε, the bound on the number of function evaluations derived in this thesis is better by a factor of $|\log \varepsilon|$ than the bound $\mathcal{O}(|\log \varepsilon|\varepsilon^{-2})$ derived in [37].

5.5 A numerical illustration

We have compared the numerical behavior of Algorithm 5.3.1 (smoothing trust-region approach) and a variant of Algorithm 5.4.1 (composite trust-region approach) on a test set suggested in [50] 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 [36] with $2 \leq n \leq 12$ and different initial points.

In the smoothing approach (`Sdfo-tr`) we used the practical trust-region implementation described in [5] for each smooth outer iteration. The implementation [5] shares some of the ideas of [32] (see Section 2.4). Unlike [32], determined quadratic models are only built when there are already $(n+1)(n+2)/2$ points evaluated. The first iteration starts with a sample set with $2n$ points of the form $x_0 \pm \mathrm{e}_i\Delta_k$) (with e_i the ith coordinate vector). Until the cardinality of the sample set reaches $(n+1)(n+2)/2$, points are never discarded from the sample set, and new trial points are always added independently of whether or not they are accepted as new iterates (in an attempt to be as greedy as possible when taking advantage of function evaluations). Models are then computed using the minimum Frobenius norm approach described in Section 2.2. Unlike [32], in the sample set update when this has reached a cardinality of $(n+1)(n+2)/2$, it is the sample point farthest away from the new iterate (instead of the current iterate) that is discarded — there is no difference if the iteration is unsuccessful. Another difference from [5] to [32] is that points that are too far from the current iterate are discarded when the trust-region radius becomes small (this can be viewed as a weak criticality condition), expecting that the next iterations will refill the sample set resulting in a similar effect as a criticality step. Notice that then the cardinality of the sample set may fall below $n+1$, the number required to build fully linear models in general, in which case the trust-region radius is not reduced. The implementation in [5] computes the minimum Frobenius models (2.18) by solving a system with (2.19) using SVD, regularizing extremely small singular values after the decomposition and before performing the backward solves (to avoid extreme ill-conditioning caused by nearly ill-poised sample sets). To improve conditioning and better assess it, the model construction (2.18) is scaled by first shifting and scaling the sample set to the unit ball as in (2.15). Finally, the trust-region subproblems are solved using the routine `trust.m` from the MATLAB Optimization Toolbox which corresponds essentially to the algorithm of Moré and Sorensen [49].

Algorithm 5.3.1 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^{-2}, which, given the initial value for μ_0, resulted in

doing four outer iterations ($k = 0, 1, 2, 4$). 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 [5] 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 (see Section 2.2) 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 [50] is given in Figure 5.1, 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}.

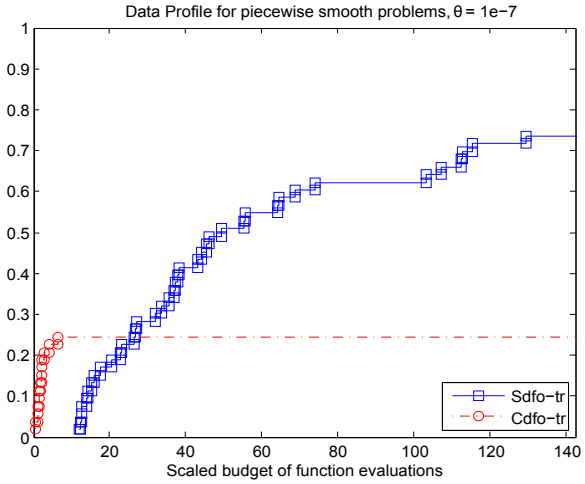

Fig. 5.1 Data profiles computed for a set of piecewise smooth problems, comparing the smoothing and composite trust-region methods.

A performance profile [30] is then given in Figure 5.2, depicting how well a method performed relatively to the other in reaching the same (scale invariant) convergence test [31], 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 [25], to ensure that we indeed measure the real ability to solve the problems.

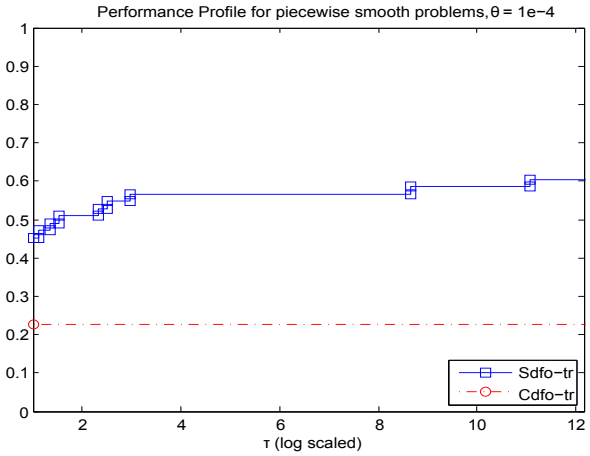

Fig. 5.2 Performance profiles computed for a set of piecewise smooth problems, in a logarithmic scale, comparing the smoothing and composite trust-region methods.

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 [34], on the same set of problems. Data and performance profiles are given in Figures 5.3 and 5.4, 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.

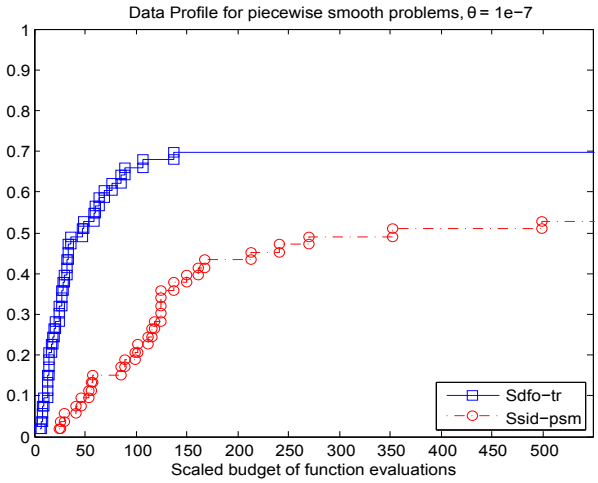

Fig. 5.3 Data profiles computed for a set of piecewise smooth problems, comparing the smoothing trust-region and direct-search methods.

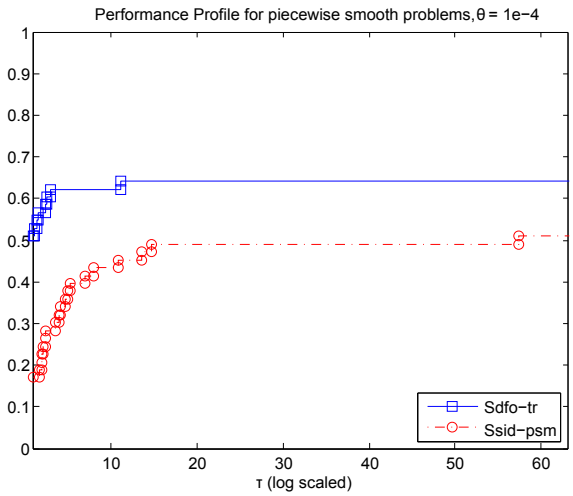

Fig. 5.4 Performance profiles computed for a set of piecewise smooth problems, in a logarithmic scale, comparing the smoothing trust-region and direct-search methods.

Chapter 6

Conclusion

This thesis presented a unified coverage of the worst case complexity 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 thesis lies in the way under which the trust-region algorithms were analyzed, individually and all together.

The analysis of WCC of this thesis can be refined along several ways. One possibility would be to establish a bound of the order of ε^{-1} when f is convex and smooth. Extension to the linearly constrained case may be doable using the methodology of this thesis, or even to the more general case where the constraints form a closed convex set.

Another topic of interest would be the investigation of the smoothing approach to determine second-order stationary points of non-smooth functions. In such a setting, one would probably consider the function continuously differentiable and smooth the second-order non-smoothness.

References

[DFO] DFO. http://www.coin-or.org/projects.html.

[2] Abramson, M. A. and Audet, C. (2006). Convergence of mesh adaptive direct search to second-order stationary points. *SIAM J. Optim.*, 17:606–619.

[3] An, L. T. H., Vaz, A. I. F., and Vicente, L. N. (2012). Optimizing radial basis functions by D.C. programming and its use in direct search for global derivative-free optimization. *TOP*, 20:190–214.

[4] Audet, C. and Dennis Jr., J. E. (2006). Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.*, 17:188–217.

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

[6] Bandeira, A. S., Scheinberg, K., and Vicente, L. N. (2014). Convergence of trust-region methods based on probabilistic models. Technical report.

[7] Bauschke, H. H., Hare, W. L., and Moursi, W. M. (2015, to appear). A derivative-free comirror algorithm for convex optimization. *Optim. Methods Softw.*

[8] Billups, S. C., Larson, J., and Graf, P. (2013). Derivative-free optimization of expensive functions with computational error using weighted regression. *SIAM J. Optim.*, 23:27–53.

[9] Booker, A. J., Dennis Jr., J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W. (1998). A rigorous framework for optimization of expensive functions by surrogates. *Structural and Multidisciplinary Optimization*, 17:1–13.

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

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

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

[13] Cartis, C., Gould, N. I. M., and Ph. L. Toint (2012a). Complexity bounds for second-order optimality in unconstrained optimization. *J. Complexity*, 28:93–108.

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

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

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

[17] Clarke, F. H. (1983). *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York. Reissued by SIAM, Philadelphia, 1990.

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

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

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

[21] Conn, A. R., Scheinberg, K., and Vicente, L. N. (2008b). Geometry of sample sets in derivative free optimization: Polynomial regression and underdetermined interpolation. *IMA J. Numer. Anal.*, 28:721–748.

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

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

[24] Curtis, F. E., Robinson, D. P., and Samadi, M. (2014). A trust-region algorithm with a worst-case iteration complexity of $\mathscr{O}(\varepsilon^{-3/2})$ for nonconvex optimization. Technical Report 14T-09, COR@L, Lehigh University.

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

[26] Davis, C. (1954). Theory of positive linear dependence. *Amer. J. Math.*, 76:733–746.

[27] Dennis, J. E., Li, S. B. B., and Tapia, R. A. (1995). A unified approach to global convergence of trust region methods for nonsmooth optimization. *Math. Program.*, 68:319–346.

[28] Dodangeh, M. and Vicente, L. N. (2015, to appear). Worst case complexity of direct search under convexity. *Math. Program.*

[29] Dodangeh, M., Vicente, L. N., and Zhang, Z. (2015, to appear). On the optimal order of worst case complexity of direct search. Technical report.

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

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

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

[33] Garmanjani, R., Júdice, D., and Vicente, L. N. (2015). Trust-region methods without using derivatives: Worst case complexity and the non-smooth case. Technical Report 15-03, Dept. Mathematics, Univ. Coimbra.

[34] Garmanjani, R. and Vicente, L. N. (2013). Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization. *IMA Journal of Numerical Analysis*, 33:1008–1028.

[35] Gould, N. I. M. and Ph. L. Toint (2010). Nonlinear programming without a penalty function or a filter. *Math. Program.*, 122:155–196.

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

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

[38] Gratton, S., Ph. L. Toint, and Troeltzch, A. (2011). An active-set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optim. Methods Softw.*, 21:873–894.

[39] Gratton, S., Royer, C. W., and Vicente, L. N. (2015). A second-order globally convergent direct-search method and its worst-case complexity. Technical Report 15-27, Dept. Mathematics, Univ. Coimbra.

[40] Gratton, S., Royer, C. W., Vicente, L. N., and Zhang, Z. (2015, to appear). Direct search based on probabilistic descent. Technical report.

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

[42] Gratton, S. and Vicente, L. N. (2014). A surrogate management framework using rigorous trust-region steps. *Optim. Methods Softw.*, 29:10–23.

[43] Gumma, E. A. E., Hashim, M. H. A., and Ali, M. M. (2014). A derivative-free algorithm for linearly constrained optimization problems. *Comput. Optim. Appl.*, 57:599–621.

[44] Hare, W. and Nutini, J. (2013). A derivative-free approximate gradient sampling algorithm for finite minimax problems. *Comput. Optim. Appl.*, 56:1–38.

[45] Kiwiel, K. C. (2010). A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization. *SIAM J. Optim.*, 20:1983–1994.

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

[47] Ph. R. Sampaio and Ph. L. Toint (2015). A derivative-free trust-funnel method for equality-constrained nonlinear optimization. Technical report.

[48] Moré, J. J. (1983). Recent developments in algorithms and software for trust region methods. *Math. Program.*, The state of the art:258–287.

[49] Moré, J. J. and Sorensen, D. C. (1983). Computing a trust region step. *SIAM J. Sci. Comput.*, 4:553–572.

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

[51] Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Comput. J.*, 7:308–313.

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

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

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

[55] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, 2nd edition.

[56] Ortega, J. M. and Rheinboldt, W. C. (1970). *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York.

[57] Powell, M. J. D. (2003). On trust region methods for unconstrained minimization without derivatives. *Math. Program.*, 97:605–623.

[58] Powell, M. J. D. (2004). Least Frobenius norm updating of quadratic models that satisfy interpolation conditions. *Math. Program.*, 100:183–215.

[59] Powell, M. J. D. (2008). Developments of NEWUOA for minimization without derivatives. *IMA J. Numer. Anal.*, 28:649–664. http://en.wikipedia.org/wiki/NEWUOA.

[60] Powell, M. J. D. (2009). The BOBYQA algorithm for bound constrained optimization without derivatives. Technical Report DAMTP 2009/NA06, University of Cambridge. http://en.wikipedia.org/wiki/BOBYQA.

[61] Powell, M. J. D. (2014). On fast trust region methods for quadratic models with linear constraints. Technical Report DAMTP 2014/NA02, University of Cambridge. http://en.wikipedia.org/wiki/LINCOA.

[62] Rockafellar, R. T. and Wets, R. J.-B. (1998). *Variational Analysis*. Springer, Berlin.

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

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

[65] Vicente, L. N. and Custódio, A. L. (2012). Analysis of direct searches for discontinuous functions. *Math. Program.*, 133:299–325.

[66] Wild, S. M., Regis, R. G., and Shoemaker, C. A. (2008). ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM J. Sci. Comput.*, 30:3197–3219.

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

[68] Yuan, Y. (1998). An example of non-convergence of trust region algorithms. *Advances in Nonlinear Programming*, ed., Kluwer Academic:205–215.

[69] Yuan, Y. (2014). Recent advances in trust region methods.

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