The Clarke Derivative and Set-Valued Mappings in the Numerical Optimization of Non-Smooth, Noisy Functions

Andreas Krahnke

Master's thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mathematics

Dr. Ekkehard W. Sachs, Chair
Dr. Robert C. Rogers
Dr. Belinda B. King
Dr. Martin V. Day

April 30, 2001
Blacksburg, Virginia

Keywords: Non-smooth Optimization, Noisy Objective Function, Clarke Derivative, Set-Valued Mapping, Implicit Filtering, Pattern Search

Copyright 2001, Andreas Krahnke

The Clarke Derivative and Set-Valued Mappings in the Numerical Optimization of Non-Smooth, Noisy Functions

Andreas Krahnke

(ABSTRACT)

In this work we present a new tool for the convergence analysis of numerical optimization methods. It is based on the concepts of the Clarke derivative and set-valued mappings. Our goul is to apply this tool to minimization problems with non-smooth and noisy objective functions.

After deriving a necessary condition for minimizers of such functions, we examine two unconstrained optimization routines. First, we prove new convergence theorems for Implicit Filtering and General Pattern Search. Then we show how these results can be used in practice, by executing some numerical computations.

Acknowledgements

Besides the efforts I put in writing up this thesis, many other people were involved in this project and contributed to its success in various direct and indirect ways. I appreciate their help and I am happy that I had the opportunity to learn from and work with them.

First of all, I would like to thank Dr. Sachs for being chair of my committee and my advisor. He especially helped me finding a topic and getting started. But he also encouraged me all of the time and took care that I stay on the right track.

I also appreciate the support of Dr. King. She guided me through the all the necessary paper work, which would have killed me otherwise. Her comments and suggestions for this final version were numerous. I guess she had a very hard and long time coping with my "German-English style". However, she did a wonderful job. Special thanks for the pumpkin-carving session and all the other invitations. I will keep these events in my mind forever.

Thanks to Dr. Rogers. Besides his work as Graduate Programm Director, he influenced me through his functional analysis lectures and his impressive presentations on several other occasions. In addition, he supported me whenever I needed it.

I have to say thank you to Dr. Battermann for proof-reading most of this work. The discussions with her were productive and stimulating. I do not know what I would have done without her mental support and motivation.

These are the people that were involved in this thesis directly, but there is more to say about my time at Virginia Polytechnic Institute and State University. It needed much more to provide the right framework for the dream of a Master of Science degree to actually come true. For example, I have never experienced such a friendly and helpful staff as the one in the Math-Office. The same holds true for the part of the faculty that I had the pleasure to meet and work with. It was also fun to experience my fellow graduate students. They always cheered me up and made the sun shine a little brighter every day. Not to forget the rest of the "German crew", Nadine and Jan. I will miss our math and non-math related discussions at lunch.

Last but definitely not least I want to mention Dr. Day. His lectures on the mathematics of financial derivatives are amazing. Always one eye on the correct mathematical concepts and the other on the well-motivated application, he was able to provide insights into this matter. I truely appreciate his ability to communicate abstract ideas and to understand a students problem. His door seems to be always open and the only thing I regret is that I did not take the time to use this opportunity more often.

Hopefully, I did not forget to mention anybody. If this should be the case, I do apologize. It did not happened by intention, but anybody makes mistakes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Every math student knows that the derivative f' of a function $f : \mathbb{R} \to \mathbb{R}$ at a point $x \in \mathbb{R}$ is given by

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \; .$$

This concept is fundamental for the analysis of functions. In particular it provides the classical necessary condition for a critical point in optimization.

However, in the applications of contemporary optimization this concept is not always appropriate. If we want to use the first order necessary condition, we have to require that the function under discussion actually has a derivative. Recent research is concerned with problems, that do not have this property. In Chapter 2 we present several alternative concepts of generalized directional derivatives, that allow to establish an analysis for non-smooth functions. There we also justify why we focus in the subsequent chapters on the so called Clarke derivative a term that was introduced by Frank H. Clarke in the seventies.

But there is another problem one encounters frequently in the complex applications nowadays. Values of the objective function we want to optimize cannot be calculated exactly. In Chapter 3 we introduce a way to take this fact into account. Instead of looking at a single-valued function, we examine a set-valued mapping surrounding this function. We show how a minimizer of such a mapping can be characterized. In addition we present a necessary condition for it, that involves a set-valued version of the Clarke derivative. We are aiming at an application of this tool in the convergence analysis of numerical optimization algorithms.

Before we can pursue this goal, we have to provide the necessary background in numerical analysis. Chapter 4 comprises the ideas behind the algorithms we want to explore. Namely we take a closer look at Implicit Filtering and a general Pattern Search method later.

We examine the former in Chapter 5. There we show how the existing convergence results can be generalized from functions that are at least smooth in a whole region to functions that are only locally Lipschitz continuous. Also the effect of inexact function values is covered in this exposition.

In contrast to Implicit Filtering, there has been done some research in connection with Pattern Search methods that aims for a non-smooth analysis using the Clarke derivative. We extend and complete this development by taking advantage of the necessary condition of Chapter 3. This is part of Chapter 6. In addition we explicitly incorporate the idea of inexact function evaluations in our analysis.

The two algorithms we discussed theoretically are examined in practice in Chapter 7. We use MATLAB implementations of them for the optimization of a model-function that is a random perturbation of a locally Lipschitzian function. The previously deduced theory helps to adjust the parameters for the particular minimization problem. The MATLAB files used are listed in the appendix.

We finish our elaborations with some concluding remarks about the current work and possible future projects in Chapter 8.

1.1 Notation

Before any confusion can arise, we address some notational issues. Most of the symbols used are defined and explained when we need them. However, there are some fundamental conventions we want to clarify at the very beginning. From this common basis we can then start to develop our theory.

Since throughout this work we will encounter point-valued functions and set-valued mappings simultaneously, we have to distinguish them somehow. In general we denote functions by small letters, e.g. $f : \mathbb{R}^n \to \mathbb{R}$. In contrast to this we use capital letters, for example $F : \mathbb{R}^n \to 2^{\mathbb{R}}$, for set-valued mappings. Analogously we name single elements $x \in \mathbb{R}^n$ by small letters, whereas whole sets are capitalized, e.g. $S \subset \mathbb{R}^n$. Especially, we denote the i^{th} canonical basis vector in $\mathbb{R}^n$ by e_i and use O for the origin of $\mathbb{R}^n$ in the case $n \geq 2$. We follow common notation by writing ∂S for the boundary of a set $S \in \mathbb{R}^n$.

We also need several norms in this text. For a vector $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ we denote the Euclidean norm by

$$\|x\| := \|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}\ .$$

In addition we use the following abbreviation for the supremum-norm for real-valued functions f that are defined on a set $S \subset \mathbb{R}^n$

$$\|f\|_S := \sup_{x \in S} |f(x)|\ .$$

Crucial for the understanding of the Clarke derivative is the notation used for generalized limits. We postpone the rigorous introduction of this concept until Chapter 2 where this idea is needed first. But before going too much into the details, we give a brief description of the main matter and important results of this work.

1.2 Scope

This work is concerned with two main issues. Recent research in optimization tries on the one hand to analyze non-smooth objective functions, because this problem is encountered in modern applied mathematics. On the other hand it is of interest to allow inexact function evaluations. The need for simplified and thus inexact models is, for example, one of many sources of inaccuracy and a reason for this interest.

As examples, we present and examine two algorithms that are subject of recent research. Implicit Filtering was especially designed to handle the problem of inaccurate function evaluations about ten years ago, compare [7], [15]. Since this time its usefulness was explored for numerous applications. Some of them can be found in [4], [6], [14]. Another method, which is employed in particular for the optimization of non-smooth objectives, is General Pattern Search (GPS). There exist various algorithms belonging to this class of optimization routines. Some of them are known for some time, like [17]. However, contemporary research is still concerned with the convergence analysis of this method, see [20], [26]. Especially, the case of non-smooth objective functions is investigated at present, compare [2], [3].

Within this work we suggest an analysis that takes both problems into account. It can handle non-smoothness and inexactness simultaneously. We show how this tool is applied to produce new results for the unconstrained versions of Implicit Filtering as well as Pattern Search. The results are very promising and motivate possible further research in this direction.

1.2.1 The Problem of Non-smooth Objective Functions

First, we discuss several possibilities to generalize the concept of a directional derivative to a larger class of functions in Chapter 2. For this purpose we have to define broader notions of tangent vectors. If C is a nonempty subset of $\mathbb{R}^n$ and $x \in \partial C$, then a vector $t \in \mathbb{R}^n$ is called a

- Bouligand tangent vector to C at x, if $\exists \, \{h_k\}_{k=1}^{\infty} \subset \mathbb{R}^+ \, , \; \{s_k\}_{k=1}^{\infty} \subset \mathbb{R}^n$ such that $h_k \downarrow 0 \, , \; s_k \to t$ and $\forall \, k \in \mathbb{N} : \;\; x + h_k s_k \in C \, ,$
- Clarke tangent vector to C at x, if $\forall \{h_k\}_{k=1}^{\infty} \subset \mathbb{R}^+ \, , \; \{x_k\}_{k=1}^{\infty} \subset C$ such that $h_k \downarrow 0 \, , \; x_k \to x \quad \exists \, \{s_k\}_{k=1}^{\infty} \subset \mathbb{R}^n$ with $s_k \to t$ and $\forall \, k \in \mathbb{N} : \;\; x_k + h_k s_k \in C \, ,$
- hypertangent vector to C at x, if $\forall \{h_k\}_{k=1}^{\infty} \subset \mathbb{R}^+ \, , \; \{x_k\}_{k=1}^{\infty} \subset C$ such that $h_k \downarrow 0 \, , \; x_k \to x \quad \exists \, N \in \mathbb{N}$ such that $\forall \, k \geq N : \;\; x_k + h_k t \in C \, .$

The entire sets of those vectors are called Bouligand contingent cone, Clarke tangent cone and hypertangent cone and are denoted by $K(C,x)$, $T(C,x)$ and $H(C,x)$, respectively. Since the Clarke tangent cone has the most interesting properties, we focus subsequently on the so-called Clarke derivative. It is induced by this cone in a natural way as we show in Section 2.3. For any function $f : \mathbb{R}^n \to \mathbb{R}$ and any point $x \in \mathbb{R}^n$ this generalized directional derivative, denoted by $Df(x,\cdot)$, exists and satisfies

$$epi\big(Df(x,\cdot)\big) \;=\; T\big(epi(f),[x,f(x)]^T\big)\ . \tag{1.1}$$

Here $epi(g)$ denotes the epigraph of a function $g : \mathbb{R}^n \to \mathbb{R}$, compare Definition 2.3. Equation (1.1) allows to transfer the advantageous properties of the Clarke tangent cone to the Clarke derivative and could also be used as a definition. Because we are interested in this concept as a tool for the convergence analysis of optimization algorithms, we prefer an alternative definition via generalized limits. The exact meaning of the following expression is explained in Section 2.1. For f and x as mentioned above and $d \in \mathbb{R}^n$ we define

$$Df(x,d) \;:=\; \lim \sup_{\substack{x_k \to x \\ y_k \to f(x), y_k \geq f(x_k) \\ h_k \downarrow 0}} \inf_{d_k \to d} \frac{f(x_k + h_k d_k) - y_k}{h_k}\ . \tag{1.2}$$

Although this expression looks complicated and probably not very useful, the Clarke derivative inherits some nice properties from it. The function $Df(x,\cdot)$ is for example lower-semicontinuous; moreover it is sublinear. The latter property is the crucial advantage of the Clarke derivative over other generalized derivatives. It is central in the reasoning for our new convergence results. Especially, the following related inequality turns out to be very helpful. For a sublinear function $f : \mathbb{R}^n \to \mathbb{R}$ it can be established that

$$\forall\, x_i \in \mathbb{R}^n,\ \lambda_i > 0\ (i = 1,\ldots,m) : \quad f\Big(\sum_{i=1}^m \lambda_i x_i\Big) \leq \sum_{i=1}^m \lambda_i f(x_i)\ . \tag{1.3}$$

However, in order to obtain an expression for the Clarke derivative we can handle, we have to restrict our attention to functions $f : \mathbb{R}^n \to \mathbb{R}$ that are at least Lipschitz continuous around a point $x \in \mathbb{R}^n$. Then equation (1.2) simplifies to

$$Df(x,d) \;=\; \limsup_{\substack{x_k \to x \\ h_k \downarrow 0}} \frac{f(x_k + h_k d) - f(x_k)}{h_k}\ .$$

We explain how to exploit the features of this equality in a while. But we also want to incorporate the possibility of inexact function evaluations. This is the topic of Chapter 3.

1.2.2 The Problem of Inexact Function Evaluations

Inexactness is frequently modelled by adding an error-function φ to the objective f that we want to optimize. As a result we get computable function values f_c of the form, compare [19, Sec. 6.1],

$$\forall\, x \in \mathbb{R}^n : \quad f_c(x) \;=\; f(x) + \varphi(x) \;.$$

Since the so-called noise φ need not be a deterministic function, but may have randomly distributed values $\varphi(x)$, we prefer an alternative model. We think of the values $f_c(x)$ to be elements of the image set of some envelope mapping F_f surrounding f. Where for some functions $f_-, f_+ : \mathbb{R}^n \to \mathbb{R}$ satisfying $f_-(x) \le f(x) \le f_+(x)$ for all $x \in \mathbb{R}^n$, we define the set-valued envelope mapping F_f to be

$$\forall\, x \in \mathbb{R}^n : \quad F_f(x) \;:=\; \{y \in \mathbb{R} \,|\, f_-(x) \le y \le f_+(x)\} \;. \tag{1.4}$$

More details on this issue can be found in Sections 3.1 and 5.3. We transfer the concept of the Clarke derivative from the point-valued to the set-valued setting by using Equation (1.1) and Definition 3.4. We define for F_f the Clarke derivative $DF_f : \mathbb{R}^n \to \mathbb{R}$ at $x \in \mathbb{R}^n$ to be the function satisfying

$$epi\big(DF_f(x,\cdot)\big) \;=\; T\big(epi(F_f), [x, f_-(x)]^T\big) \;.$$

After verifying the existence and uniqueness of such a function in Section 3.2, we observe the equality $DF_f(x,\cdot) = Df_-(x,\cdot)$. Also, in the special case $f_- = f = f_+$, we find out that the two derivatives $DF_f(x,\cdot)$ and $Df(x,\cdot)$ coincide. Thus, the Clarke derivative for envelope mappings is an extension of the corresponding concept for functions. It enables us to analyze functions that are not smooth and additionally cannot be evaluated exactly.

In order to make use of this concept to its full extent, we present a necessary condition for a minimizer based on the Clarke derivative. This tool plays a central role in establishing a really non-smooth analysis and is based on the next theorem. It is presented and proven later in this thesis as well as the other results that are quoted throughout the remainder of this section.

Theorem.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x_* \in \mathbb{R}^n$ and F_f an envelope mapping around f given by (1.4). If the vector $z_-(x_*) = \binom{x_*}{f_-(x_*)}$ is a local minimizer of F_f and $DF_f(x_*,\cdot)$ exists, then

$$\forall\, d \in \mathbb{R}^n : \qquad DF_f(x_*, d) \ge 0 \;. \tag{1.5}$$

We note that the above holds especially for $F_f(\cdot) = \{f(\cdot)\}$ and $DF_f(x,\cdot) = Df(x,\cdot)$. This result is a generalization of the classical necessary conditions, which require a stronger notion of differentiability. The consequence of this generalization is that the new necessary condition is weaker than the classical ones. This topic is discussed in detail in Section 3.3.

1.2.3 New Convergence Results

Our new tool wants to be applied. We do so after presenting in Chapter 4 some theoretical background from numerical analysis and linear algebra. This is necessary for the proper understanding of our subsequent reasoning. Especially, the idea of a positive spanning set has to be clarified. It is a set of vectors $\{p_i\}_{i=1}^m \subset \mathbb{R}^n$ that has the property

$$\forall\, x \in \mathbb{R}^n \;\exists\; \{\lambda_i\}_{i=1}^m \subset \mathbb{R}_0^+ \;\text{ such that }\quad x = \sum_{i=1}^m \lambda_i p_i \;. \tag{1.6}$$

Our first result is a broader convergence statement for Implicit Filtering presented in Section 5.3. We basically make the same assumptions that Kelley needs in [19]. We only have to modify the finite-difference approximations used within the algorithm to obtain the following theorem, compare Theorem 5.4.

Theorem.
Let $f_-, f, f_+ : \mathbb{R}^n \to \mathbb{R}$ such that f_- is bounded below on $\mathbb{R}^n$ and let F_f be the envelope mapping around f given by (1.4). In addition let $\{h_k\}_{k=0}^\infty$ be a sequence of scales such that $h_k \downarrow 0$ and $\{x_k\}_{k=0}^\infty$ be the sequence produced by the modified Implicit Filtering algorithm using simplex gradient approximations on $S_k = S(x_k, h_k)$ and computed function values $f_c(\cdot) \in F_f(\cdot)$. If we assume that there are at most finitely many line search failures and

$$\lim_{k\to\infty} \frac{\|f_+ - f_-\|_{S_k \cup R(S_k)}}{h_k} = 0 \tag{1.7}$$

holds, then any limit point x_* of $\{x_k\}_{k=1}^\infty$, around which f is Lipschitz continuous, is a critical point of f, i.e.

$$\forall\, d \in \mathbb{R}^n : \quad Df(x_*, d) \geq 0 \;.$$

As claimed before, this is a convergence result that simultaneously treats non-smoothness and the difficulty of inexact function evaluations. It generalizes the results of [8] and [19] for a larger class of functions. For this we do not even need to impose additional prerequisites. We only have to change to a weaker necessary condition.

A similar observation can be made for General Pattern Search (GPS). There already exist propositions for this method that aim for the analysis of the non-smooth case. We first contribute to these efforts by incorporating our necessary condition (1.5) in this context. This leads to an interim result, compare Theorem 6.11, that is designed for non-smooth functions with exactly available function values. The term *refining subsequence*, which we use soon, is defined in Subsection 6.2.2.

Theorem.
Let $x_0 \in \mathbb{R}^n$ be the starting point for a GPS method and $f : \mathbb{R}^n \to \mathbb{R}$ be such that

$L(x_0) := \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ is bounded. Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of iterates produced by the GPS method with objective f and let x_* be the limit of a refining subsequence. If f is Lipschitz continuous around x_*, then

$$\forall\, d \in \mathbb{R}^n : \quad Df(x_*, d) \geq 0 \,.$$

But we can do better than that and additionally handle the problem of inexactness. For this purpose we formally surround the function of interest by an envelope mapping. By doing so we are able to prove the following, see also Theorem 6.12.

Theorem.
Let $x_0 \in \mathbb{R}^n$ be the starting point for a GPS algorithm and $f : \mathbb{R}^n \to \mathbb{R}$. Let $f_-, f_+ : \mathbb{R}^n \to \mathbb{R}$ and F_f be the envelope mapping around f given by (1.4). Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of iterates produced by the GPS method with objective function values $f_c(\cdot) \in F_f(\cdot)$. Let $L_c(x_0) := \{x \in \mathbb{R}^n \mid f_c(x) \leq f_c(x_0)\}$ be bounded and x_* be the limit of a refining subsequence $\{x_j\}_{j\in J}$ with associated positive spanning set $\{p_i\}_{i=1}^{2n}$. For $j \in J$ let denote $B_j(x_j)$ the smallest ball around x_j containing the set of vectors $\{x_j + h_j p_i\}_{i=1}^{2n}$. If f is Lipschitz continuous around x_* and

$$\lim_{\substack{j\to\infty \\ j\in J}} \frac{\|f_+ - f_-\|_{B_j(x_j)}}{h_j} = 0 \tag{1.8}$$

holds, then

$$\forall\, d \in \mathbb{R}^n : \quad Df(x_*, d) \geq 0 \,.$$

This theorem and the result we established for Implicit Filtering are both based on the same mechanism. At first we use the specifics of the algorithms to show for the computed function values and some positive spanning set $\{p_i\}_{i=1}^{m}$ that

$$\forall\, i \in \{1, \ldots, m\} : \quad \limsup_{k\to\infty} \frac{f_c(x_k + h_k p_i) - f_c(x_k)}{h_k} \geq 0 \,.$$

Then we use the assumed accuracy of the envelope mapping to get for the Lipschitzian objective f

$$\forall\, i \in \{1, \ldots, m\} : \quad Df(x_*, p_i) \geq 0 \,.$$

Finally we use the sublinearity of the Clarke derivative together with the specifics of a positive spanning set, compare (1.3) and (1.6), to conclude

$$\forall\, d \in \mathbb{R}^n : \quad Df(x_*, d) \geq 0 \,.$$

1.2.4 Implications and Conclusions

These results are not only of theoretical interest. In fact they justify the application of both algorithms to noisy, non-smooth problems. Furthermore, they can help to choose suitable parameters for a practical application. In Chapter 7 we show how especially the Equations (1.7) and (1.8) provide information for this task. We use a locally Lipschitz continuous test function for the exemplary calculations. This objective is perturbed by some randomly generated noise. Thus, the optimization routines can really be presumed to use arbitrary elements of the image set of some surrounding envelope mapping.

Even though we restrict our attention in this basic work to unconstrained minimization problems, the results we get are remarkable. They encourage further work, which can be directed into various fields of interest. First of all, it seems to be possible to extend the new analysis to the constrained setting. Of course it is also reasonable to try to use this tool to find convergence results for other algorithms as well. Another attractive task are further relaxations of the prerequisites. Maybe, one can weaken the assumption of local Lipschitz continuity. Alternatively, we could allow a higher level of noise. It is likely that we cannot guarantee convergence to a minimum under these circumstances. But it is a challenge to search for an upper bound for the difference between the practically computed and the theoretically exact minimum value. However, before we discuss these future plans any further, we have to manage the details of the current work.

Chapter 2

Generalized Derivatives

In this chapter we present generalized definitions of the derivative of a real-valued function $f : \mathbb{R}^n \to \mathbb{R}$. First we state some basic definitions in Section 2.1. Then we introduce various geometrical concepts of tangents and tangent cones in Section 2.2. After this groundwork we are well prepared to explore the various generalized ways of differentiation in Section 2.3.

By abstracting and modifying the basic idea of a tangent vector, new concepts are created. This assures the usefulness for optimization, since the tangent property of the *classical* derivative is the key for proving necessary conditions. However, not all of the concepts are equally promising. For example in general we cannot guarantee the linearity of the directional derivative, because the new concepts do not attempt to approximate the function f locally by linearization.

Later, we focus on the so-called Clarke derivative, which was introduced by Frank H. Clarke in the seventies (see [9]). Instead of linearizing f, it is convexified. By this we are able to guarantee at least sublinearity for this directional derivative. This feature is crucial for the proofs of Chapters 5 and 6. It is also the reason why we prefer this concept to the others presented in this chapter.

But other generalized derivatives are also used successfully. For example the contingent epiderivative plays an important role in set-valued optimization. It opens up the possibility to establish new necessary and sufficient conditions (see for example [16], [18], [21]). Likewise we introduce in Chapter 3 a necessary condition involving the Clarke derivative.

2.1 Basic definitions

Since all concepts we present in Section 2.3 are based on a geometric approach, we first define convex cones. This geometric construction is essential for our further elaborations, but we do not introduce much more geometric background. For an introduction to this matter we refer the interested reader to [23].

Definition 2.1.
Let $C \subset \mathbb{R}^n$ and $C \neq \emptyset$. If it has the properties

i) $\forall\, x \in C\,,\ \alpha \in \mathbb{R}_0^+ : \quad \alpha x \in C$ and

ii) $\forall\, x, y \in C : \quad x + y \in C$,

then C is called a *convex cone*. If it has in addition the property

iii) $x \in C$ and $-x \in C \quad \Rightarrow \quad x = O$,

then it is called a *pointed convex cone*. □

Of course this definition, as most of the following, could be generalized, but we refrain from doing so. This is because our application is also set in $\mathbb{R}^n$ and we do not need more than the above. Before we proceed, we take a look at an example in $\mathbb{R}^2$.

Example 2.2.
In Figure 2.1 below we see on the left a convex cone that is not pointed. In contrast to this the shaded area on the right is a pointed convex cone. □

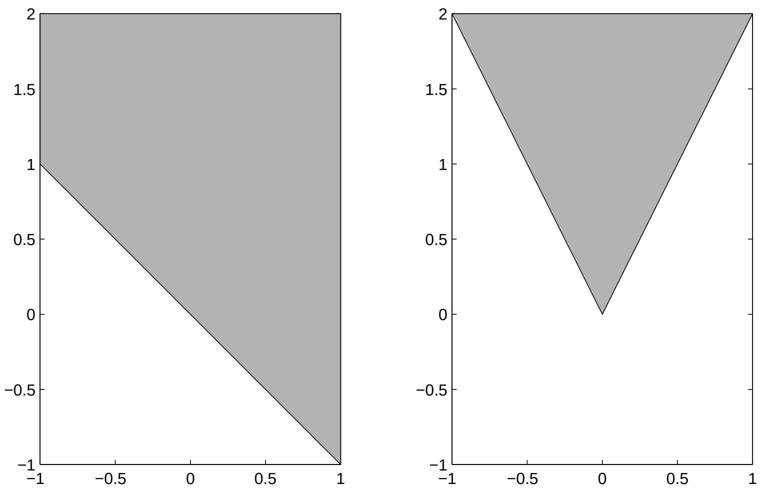

Figure 2.1: Nonpointed (left) and pointed cone (right) in $\mathbb{R}^2$

Another special set we have to talk about is the epigraph of a function. It is the connection between abstract geometry and the functions of which we want to create a derivative.

Definition 2.3.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a real-valued function. Then the set

$$epi(f) \ := \ \left\{ \binom{x}{y} \in \mathbb{R}^n \times \mathbb{R} \middle|\ y \geq f(x) \right\} \subset \mathbb{R}^{n+1}$$

is called the *epigraph* of f. □

Of course the properties of the epigraph, $epi(f)$, depend on the properties of the function f and vice versa. The next definition is an example of this correlation. We already mentioned the sublinearity of a function. Since this notion is necessary for our later proofs, we already introduce it now at the very beginning.

Definition 2.4.
A function $f : \mathbb{R}^n \to \mathbb{R}$ is called *sublinear*, if its epigraph is a convex cone containing the origin. $\square$

This definition might look not very telling. But it helps to establish Corollary 2.22 in Section 2.3, which is indispensable for our work. However, to get a better idea of what sublinearity means we mention an equivalent formulation. A function f is sublinear if and only if f is convex, satisfies $f(0) < \infty$, and for all $\lambda > 0$, $x \in \mathbb{R}^n$ we find $f(\lambda x) = \lambda f(x)$. This immediately implies Inequality (2.1), which motivates the name sublinearity and is crucial for the results of the subsequent chapters.

$$\forall\, x_i \in \mathbb{R}^n,\ \lambda_i > 0\ (i = 1, \dots, m) : \quad f\Big(\sum_{i=1}^{m} \lambda_i x_i\Big) \le \sum_{i=1}^{m} \lambda_i f(x_i)\,. \tag{2.1}$$

There is one last topic left that we have to introduce, generalized limits. In $\mathbb{R}^n$ we are used to writing for $x \in \mathbb{R}^n$ and a converging sequence $\{x_k\}_{k=1}^{\infty} \subset \mathbb{R}^n$

$$\lim_{k\to\infty} x_k = x \quad :\Leftrightarrow \quad \|x_k - x\| \to 0\ .$$

For a continuous function $f : \mathbb{R}^n \to \mathbb{R}$ and any sequence $\{x_k\}_{k=1}^{\infty} \subset \mathbb{R}^n$ converging to x we know that the sequence of function values also converges to $f(x)$ and write

$$\lim_{k\to\infty} f(x_k) = f(x) \quad :\Leftrightarrow \quad |f(x_k) - f(x)| \to 0\ .$$

We note that this limit does not depend on the choice of the sequence and define in general:

Definition 2.5.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}$, then we write in the case of existence

$$\lim_{x_k\to x} f(x_k) = y \quad :\Leftrightarrow \quad \forall\, \{x_k\}_{k=1}^{\infty} \subset \mathbb{R}^n \text{ with } x_k \to x : \ \ |f(x_k) - y| \to 0\ .$$

$\square$

At this comprehensible level we want to mention that there exists a topological equivalent to Definition 2.5.

Lemma 2.6.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}$. For $\delta > 0$ let $B_\delta(x)$ denote the ball with radius δ around x. Then the following are equivalent:

i) $\lim\limits_{x_k \to x} f(x_k) = y$,

ii) $\forall\, \varepsilon > 0\; \exists\, \delta = \delta(x,\varepsilon) > 0$ *such that* $\forall\, x' \in B_\delta(x): \;\; |f(x') - y| < \varepsilon$,

iii) $\sup\limits_{\delta>0} \inf\limits_{x' \in B_\delta(x)} f(x') = \inf\limits_{\delta>0} \sup\limits_{x' \in B_\delta(x)} f(x') = y$.

The last characterization motivates the following definition of a limes inferior and limes superior.

Definition 2.7.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}$. For $\delta > 0$ let $B_\delta(x)$ denote the ball with radius δ around x. Then we write:

$$\text{i)}\;\; \liminf_{x_k \to x} f(x_k) = y \quad :\Leftrightarrow \quad \sup_{\delta>0} \inf_{x' \in B_\delta(x)} f(x') = y \quad \Leftrightarrow \quad \lim_{\delta \to 0} \inf_{x' \in B_\delta(x)} f(x') = y \;,$$

$$\text{ii)}\;\; \limsup_{x_k \to x} f(x_k) = y \quad :\Leftrightarrow \quad \inf_{\delta>0} \sup_{x' \in B_\delta(x)} f(x') = y \quad \Leftrightarrow \quad \lim_{\delta \to 0} \sup_{x' \in B_\delta(x)} f(x') = y \;.$$

□

Note that the last equivalences are true, because

$$\inf_{x' \in B_\delta(x)} f(x') \quad \text{and} \quad \sup_{x' \in B_\delta(x)} f(x')$$

are monotonic in δ. Also be aware of the following fact

$$\liminf_{x_k \to x} f(x_k) = y \quad \not\Leftrightarrow \quad \forall\, \{x_k\}_{k=1}^\infty \subset \mathbb{R}^n \text{ with } \; x_k \to x: \; \liminf_{k \to \infty} f(x_k) = y \;.$$

Obviously the "$\Leftarrow$" direction is true. We give an example to show that the other implication is wrong. This is why Definition 2.7 is based on topological concepts and not on sequential limits.

Example 2.8.
Let $f : \mathbb{R} \to \mathbb{R}$ be defined by $f(x) := \delta_{\mathbb{R}^+}(x)$, the indicator function of $\mathbb{R}^+$, compare Figure 2.2 on the next page. Clearly we find

$$\liminf_{x_k \to 0} f(x_k) = 0.$$

But for $x_k = \frac{1}{k}$ $(k \in \mathbb{N})$ we observe $x_k \to 0$ and

$$\liminf_{k \to \infty} f(x_k) = \lim_{k \to \infty} f(x_k) = 1 \;.$$

□

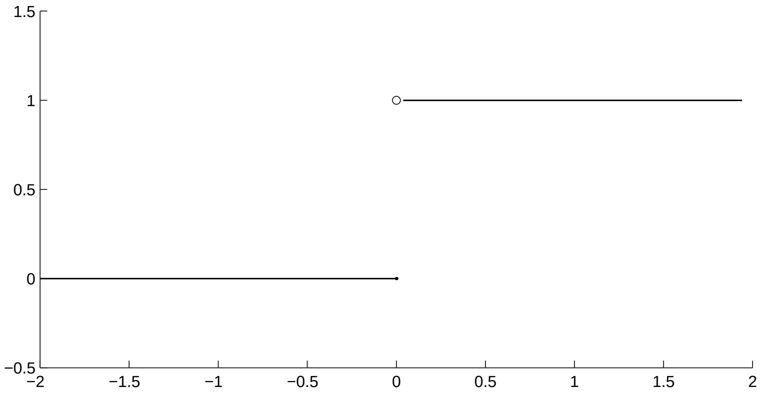

Figure 2.2: Graph of $f(x) = \delta_{\mathbb{R}^+}(x)$

To be able to talk about generalized derivatives in Section 2.3 in a rigorous and unambiguous way, we have to extend our definitions in the following way, compare [24].

Definition 2.9.
Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and $z \in \mathbb{R}$. For $\delta > 0$ let $B_\delta(x)$ and $B_\delta(y)$ denote the balls with radius δ around x and y, respectively. Then we write

$$\text{i)}\ \limsup_{x_k \to x} \inf_{y_k \to y} f(x_k, y_k) = z \quad :\Leftrightarrow \quad \sup_{\delta_1 > 0} \inf_{\delta_2 > 0} \sup_{x' \in B_{\delta_2}(x)} \inf_{y' \in B_{\delta_1}(y)} f(x', y') = z$$

$$\Leftrightarrow \quad \lim_{\delta \to 0} \sup_{x' \in B_\delta(x)} \inf_{y' \in B_\delta(y)} f(x', y') = z\ ,$$

$$\text{ii)}\ \liminf_{x_k \to x} \sup_{y_k \to y} f(x_k, y_k) = z \quad :\Leftrightarrow \quad \inf_{\delta_1 > 0} \sup_{\delta_2 > 0} \inf_{x' \in B_{\delta_2}(x)} \sup_{y' \in B_{\delta_1}(y)} f(x', y') = z\ .$$

$$\Leftrightarrow \quad \lim_{\delta \to 0} \inf_{x' \in B_\delta(x)} \sup_{y' \in B_\delta(y)} f(x', y') = z$$

□

In Section 2.3, slight variations of the above notation occur. But once the above definitions are understood, it is intuitive and obvious how to modify them correctly. For example we would define for a function $f : \mathbb{R} \to \mathbb{R}$

$$\liminf_{h_k \downarrow 0} f(h_k) = y \quad :\Leftrightarrow \quad \lim_{\delta \to 0} \inf_{h' \in (0,\delta)} f(h') = y\ .$$

Before we use these generalized limits for the definition of derivatives, we present the geometric concepts that they are based on. Namely these are tangents and tangent cones.

2.2 Tangents and Tangent Cones

Suppose we are given a ball B in $\mathbb{R}^2$. Suppose in addition that we are also given a point x on its boundary. Then we can think of the tangent line to B at x as the limit of secant lines. This idea is displayed in Figure 2.3 below.

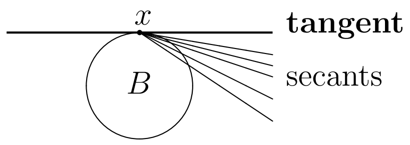

Figure 2.3: The tangent to a ball as the limit of secants

If we abstract this two-dimensional picture to $\mathbb{R}^n$ and think of secants as vectors that when added to x, lie within the set B, we are close to a definition for general tangent vectors. They can be perceived as a limit of these secants. What the set of all those tangents looks like and what properties it possesses, depends on how we take the limit of the secant vectors.

2.2.1 The Contingent Cone

At first we introduce a notion of tangent vectors, which establishes that set of all such vectors is closed.

Definition 2.10.
Let $C \subset \mathbb{R}^n$ be a nonempty set and $x \in \partial C$. Then a vector $t \in \mathbb{R}^n$ is called a *Bouligand tangent vector to C at x*, if there exist sequences $\{h_k\}_{k=1}^{\infty} \subset \mathbb{R}^+$ and $\{s_k\}_{k=1}^{\infty} \subset \mathbb{R}^n$ such that

$$h_k \downarrow 0 \quad , \quad s_k \to t \quad \text{and} \quad \forall\, k \in \mathbb{N}: \;\; x + h_k\, s_k \in C \;.$$

The set of all those tangent vectors t is called the *(Bouligand) contingent cone to C at x* and we denote it by $K(C,x)$. □

In the literature, x is often not required to be on the boundary of C (see e.g. [25]). We make this restriction, since it is the only interesting case for us. Note that otherwise the contingent cone would equal the whole of $\mathbb{R}^n$, if x were in the interior of C. We now describe some of the characteristics of $K(C,x)$ by looking at a simple example in $\mathbb{R}^2$.

Example 2.11.
Figure 2.4 on the next page comprises the graph of the function $f(x) := 1 - e^{|x|}$ (black line) and the contingent cone to its epigraph at the origin O (shaded in grey). Notice that the boundary of the grey area belongs to the contingent cone as well. Thus the latter is a closed set. Also the origin belongs to $K(epi(f), O)$. □

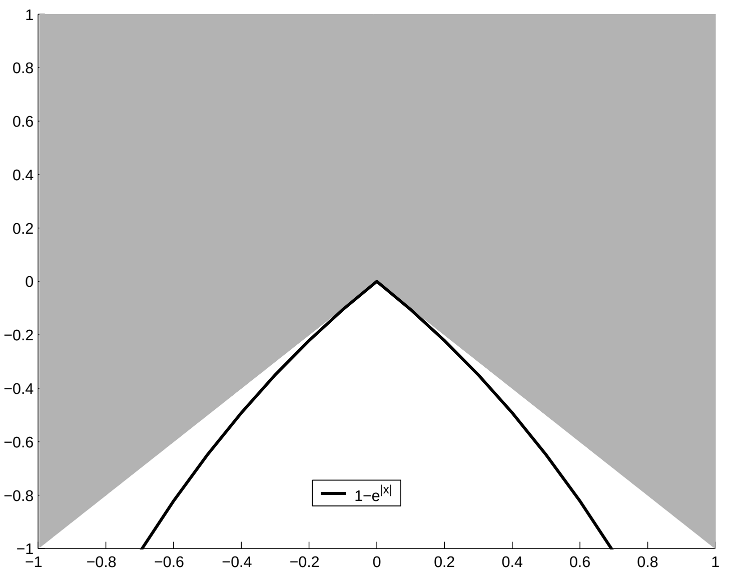

Figure 2.4: Contingent cone to the epigraph of $f(x) := 1 - e^{|x|}$ at O

In fact it can be shown that $K(C,x)$ is always a closed cone containing the origin. Figure 2.4 also demonstrates that in general the contingent cone fails to be convex. This property, which is desirable for optimization purposes, cannot in general be assumed. Of course, we want to find a set of tangents that is convex in every case. We soon examine the Clarke tangent cone. It does exhibit this property.

We want to mention the existence of other formulations for the contingent cone. Our definition is based on the limits of sequences. For example there also exists a topological equivalent which is given in (2.2). There $B_\delta(t)$ denotes the ball with radius δ around t.

$$K(C,x) = \{\, t \in \mathbb{R}^n \,|\, \forall\, \delta > 0 \;\; \exists\, h' \in (0,\delta), s' \in B_\delta(t) : \;\; x + h's' \in C\}. \tag{2.2}$$

Depending on the context in which one uses the contingent cone, one might prefer the sequential or the topological definition. The same holds for the other cones to which we turn our attention now.

2.2.2 The Clarke Tangent Cone

Next we take a look at the construction that is most interesting and valuable for us. The Clarke tangent cone convexifies any nonempty set at one of its limit points.

Definition 2.12.
Let $C \subset \mathbb{R}^n$ be a nonempty set and $x \in \partial C$. Then a vector $t \in \mathbb{R}^n$ is called a *Clarke tangent vector to C at x*, if for all $\{h_k\}_{k=1}^\infty \subset \mathbb{R}^+$, $\{x_k\}_{k=1}^\infty \subset C$ such that

$$h_k \downarrow 0 \qquad \text{and} \qquad x_k \to x$$

there exists a sequence $\{s_k\}_{k=1}^{\infty} \subset \mathbb{R}^n$ that satisfies

$$s_k \to t \qquad \text{and} \qquad \forall k \in \mathbb{N} :\ x_k + h_k\, s_k \in C\ .$$

The set of all those tangent vectors is called the *Clarke tangent cone to C at x* and denoted by $T(C,x)$. □

Obviously, the two definitions we gave so far are closely related. Going more into detail we soon find out that the Clarke tangent cone is always a subset of the contingent cone:

Let t be a Clarke tangent vector to C at x, i.e.

$$\forall\, \{h_k\}_{k=1}^{\infty}, h_k \downarrow 0 \quad \forall\, \{x_k\}_{k=1}^{\infty} \in C, x_k \to x \quad \exists\, \{s_k\}_{k=1}^{\infty} \in \mathbb{R}^n, s_k \to t \quad \text{and}$$
$$\forall\, k \in \mathbb{N} : \quad x_k + h_k\, s_k \in C\ .$$

In particular if we pick for all $k \in \mathbb{N} :\ x_k = x$, then we know that

$$\exists\, \{h_k\}_{k=1}^{\infty}, h_k \downarrow 0 \quad , \exists\, \{s_k\}_{k=1}^{\infty} \in \mathbb{R}^n, s_k \to t \ \text{ and } \ \forall\, k \in \mathbb{N} : \quad x + h_k\, s_k \in C\ .$$

Thus t is by definition a Bouligand tangent vector.

We again want to look at an example in $\mathbb{R}^2$ to get a better idea of what the given definition means.

Example 2.13.
To make our two examples comparable, we choose the same set C – the epigraph of the function $f(x) := 1 - e^{|x|}$ (black line) – and the same point x, i.e. the origin.

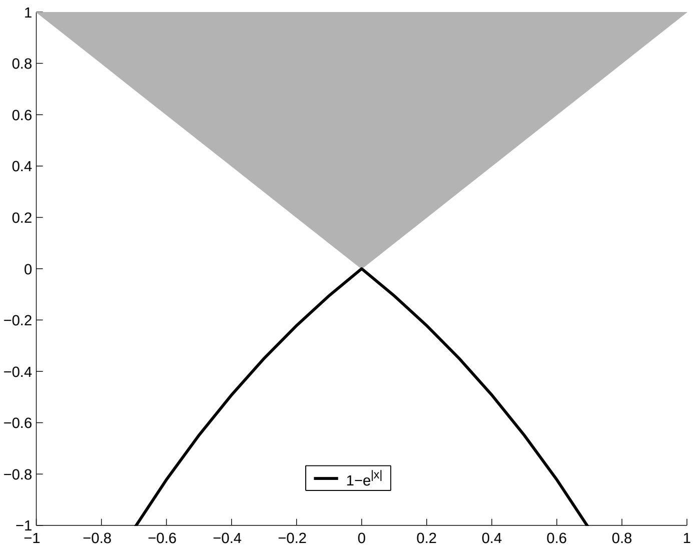

Figure 2.5: Clarke tangent cone to the epigraph of $f(x) := 1 - e^{|x|}$ at O

We observe the just proven relation $T(C,x) \subset K(C,x)$ by comparing Figures 2.4 and 2.5. Again the boundary belongs to the gray cone, which makes $T(epi(f), O)$ a closed set containing the origin. In the above graph the Clarke tangent cone is clearly convex, too. $\square$

However, convexity is not so easy to deduce in general from the definition via sequential limits. That is why we now examine some alternatives, first of all the topological one. It can be shown that

$$\begin{aligned} T(C,x) \; &= \; \{\, t \in \mathbb{R}^n \,|\, \forall\, \delta_1 > 0 \;\; \exists\, \delta_2 > 0 \;\; s.t. \;\; \forall\, x' \in C \cap B_{\delta_2}(x), h' \in (0, \delta_2) \\ &\qquad \exists\, s' \in B_{\delta_1}(t) : \;\; x' + h's' \in C \} \, . \end{aligned} \tag{2.3}$$

We use here $B_\delta(x)$ in the same way as we did in (2.2). Again this shows the inclusion of the Clarke tangent cone in the Bouligand tangent cone. But to see where the convexity comes from, we have to consider yet another, indirect definition via normals and polar cones. A well motivated approach to this can be found in the book of Clarke [10, Ch.1]. We outline the basic ideas.

Instead of taking tangent vectors as a starting point, one can also begin with an abstraction of what it means for a vector to be perpendicular to a set. By using the projection theorem we define in very general terms for any convex subset C of a Hilbert space $(H, \langle\cdot,\cdot\rangle)$ and any point p outside C a unique point x_p on the boundary of C closest to p. We then call $p - x_p$ perpendicular to C at x_p.

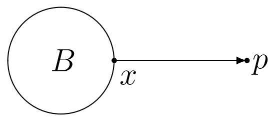

Figure 2.6: Perpendicular vector to the ball B at x

However, for an arbitrary set C and $x \in \partial C$ there may be multiple $p \notin C$ such that x is closest to p in C. We call the collection of all perpendicular vectors at a point $x \in \partial C$ and their nonnegative multiples the set of *proximal normal vectors*. Note that points in the interior of C only have O as trivial proximal normal. This justifies once more our focus on points $x \in \partial C$. For them the proximal normal vectors are used to define a closed convex cone containing the origin. It is called the *normal cone* and usually denote by $N(C,x)$. Finally we can apply polarity to define $T(C,x)$ to be the polar cone of $N(C,x)$. Since the polar of a closed convex cone is again closed and convex, the desired property of $T(C,x)$ is built into this indirect definition. One may ask why this definition coincides with the one given at the beginning. We do not prove this fact, but only refer to [9, Prop. 3.7].

Having Definition 2.4 in mind, it is clear what the convexity of the Clarke tangent cone will be used for. It will establish sublinearity for the Clarke derivative in Section 2.3.

2.2.3 The Hypertangent Cone

Before we advance to the next section, we want to present a third kind of tangent cone.

Definition 2.14.
Let $C \subset \mathbb{R}^n$ be a nonempty set and $x \in \partial C$. Then a vector $t \in \mathbb{R}^n$ is called a *hypertangent vector to C at x*, if for all $\{h_k\}_{k=1}^{\infty} \subset \mathbb{R}^+$, $\{x_k\}_{k=1}^{\infty} \subset C$ such that

$$h_k \downarrow 0 \qquad \text{and} \qquad x_k \to x$$

there exists a positive integer $N \in \mathbb{N}$ such that

$$\forall\, k \geq N : \quad x_k + h_k\, t \in C\ .$$

The set of all those tangent vectors is called the *hypertangent cone to C at x* and denoted by $H(C,x)$. □

Directly by the definition we can see that this is an even more restrictive characterization of a tangent vector than the one of Clarke. The before arbitrary sequence $\{s_k\}_{k=1}^{\infty}$ converging to the tangent vector t is now replaced by a sequence that is eventually equal to t. This implies: $H(C,x) \subset T(C,x)$. This difference is apparent in our example below, too.

Example 2.15.
Again we plotted the function $f(x) := 1 - e^{|x|}$. This time we added the hypertangent cone $H(epi(f), O)$. Note that now the boundary does no longer belong to the cone.

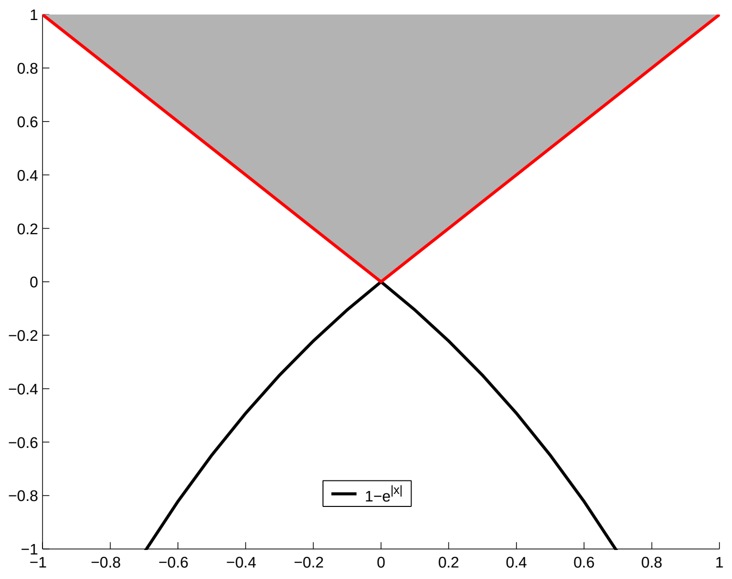

Figure 2.7: The function $f(x) := 1 - e^{|x|}$ and $H(epi(f), O)$

Consequently $H(C,x)$ in general can fail to be a closed set. However, it can be shown that every hypertangent cone is a convex cone. □

Another way of thinking about hypertangents and what their restrictive definition means is suggested by the topological expression

$$H(C,x) = \{\, t \in \mathbb{R}^n \,|\, \exists\, \delta > 0 \;\; s.t. \;\; \forall\, x' \in C \cap B_\delta(x), h' \in (0,\delta) : \;\; x' + h't \in C\}.$$

Thus nonzero vectors in $H(C,x)$ are directions for *"moving uniformly"* from x into the set C.

Before we conclude this section about tangent cones, we summarize the presented results in a small table.

type of cone	$K(C,x) \;\; \supset$	$T(C,x) \;\; \supset$	$H(C,x)$
closed in general	yes	yes	no
convex in general	no	yes	yes

Table 2.1: The different tangent cones and their properties

Table 2.1 confirms once more the outstanding properties of the Clarke tangent cone. It seems most promising for the definition of a generalized derivative that allows to state and prove at least a useful necessary condition.

2.3 Derivative Concepts

In this section we define two types of generalized derivatives. Although we have already picked the Clarke derivative as our favorite, it makes sense to present the contingent epiderivative as an additional concept. Not only that it gained much attention in set-valued optimization (see [16], [18], [21]), but it also makes a pattern for the definition of generalized derivatives apparent. Compared with the Clarke derivative it suggests how to define similar concepts. We briefly discuss the idea that any kind of tangent cone seems to induce a more or less useful generalized derivative at the end of Subsection 2.3.2.

2.3.1 The Contingent Epiderivative

One of the more useful concepts is definitely the contingent epiderivative.

Definition 2.16.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function and $x, d \in \mathbb{R}^n$, then the *contingent epiderivative Ef at x with respect to the direction d* is defined as

$$Ef(x,d) := \liminf_{\substack{d_k \to d \\ h_k \downarrow 0}} \frac{f(x + h_k d_k) - f(x)}{h_k} \; . \tag{2.4}$$

□

When reading Definition 2.16, one should be aware of the exact meaning of the used *liminf*-expression. We introduced this kind of limit in Definition 2.7. It guarantees that the function $Ef(x,\cdot)$ for any $f : \mathbb{R}^n \to \mathbb{R}$ is well defined. However, it is not necessarily real-valued. There are instances where the contingent epiderivative is an extended real-valued function with values in $\mathbb{R} \cup \{\pm\infty\}$.

Note that this definition is a generalization of the Fréchet derivative f' of a function $f : \mathbb{R}^n \to \mathbb{R}$ at the point $x \in \mathbb{R}^n$ in the direction $d \in \mathbb{R}^n$, which is in the case of existence given by

$$f'(x,d) = \lim_{\substack{d_k \to d \\ h_k \downarrow 0}} \frac{f(x + h_k d_k) - f(x)}{h_k} \ .$$

Further on we also refer to the function $f'(x,\cdot)$ as the *classical derivative*. It is well known that $f'(x,\cdot)$ is a bounded linear functional. Since $\mathbb{R}^n$ is a Hilbert space, we find by the Riesz representation theorem

$$\exists\, g \in \mathbb{R}^n \text{ such that } \forall\, d \in \mathbb{R}^n : \quad f'(x,d) = d^T g \ .$$

The vector g is called the gradient of f at x and usually denoted by $\nabla f(x)$. We cannot guarantee anything like linearity for the contingent epiderivative. This is its main failing and the reason why we prefer the Clarke derivative. For the latter we can prove at least sublinearity.

But to get a better notion of the contingent epiderivative first, we state the following basic fact.

Lemma 2.17.
Let $f : \mathbb{R}^n \to \mathbb{R}$ *and* $x \in \mathbb{R}^n$. *Define* $z(x) := \binom{x}{f(x)}$, *then the following equation holds*

$$\mathit{epi}\,(Ef(x,\cdot)) = K\,(\mathit{epi}(f), z(x)) \ . \tag{2.5}$$

We omit the proof, because this result is only presented for the understanding of the concept and not used later on. It can be found in [22, Sec.3,Thm.1]. This theorem allows us to quickly explore the following helpful example.

Example 2.18.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be defined by $f(x) := 1 - e^{|x|}$. Obviously f is not differentiable at $x = O$ in the classical sense. However, we have introduced this function already in Section 2.2. We were even able to draw $K\,(\mathit{epi}(f), O)$ in Figure 2.4. From there we immediately observe – using Theorem 2.17 – that $Ef(O,d) = -|d|$. □

The theorem above can also be seen as an alternative definition of the contingent epiderivative. It is then defined to be the function $Ef(x,\cdot) : \mathbb{R}^n \to \mathbb{R}$ that satisfies (2.5). This is an appropriate way to transfer the idea of a derivative from functions to set-valued mappings, as we see in Chapter 3. This has already been done for the contingent epiderivative in for example [16],[18]. We now look at the Clarke derivative and find a similar theorem. This motivates a general way of defining generalized derivatives by tangent cones.

2.3.2 The Clarke Derivative

As in the previous subsection, we introduce a new kind of derivative via a sequential limit. After that we give an alternative definition and discuss the advantageous properties of the Clarke derivative. Finally, we suppose a way to define other types of generalized derivatives.

Definition 2.19.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function and $x, d \in \mathbb{R}^n$, then the *Clarke derivative Df at x with respect to the direction d* is defined as

$$Df(x,d) \;:=\; \lim \sup_{\substack{x_k \to x \\ y_k \to f(x), y_k \geq f(x_k) \\ h_k \downarrow 0}} \inf_{d_k \to d} \frac{f(x_k + h_k d_k) - y_k}{h_k} \,. \tag{2.6}$$

□

This somewhat unhandy expression can be simplified under additional assumptions. We make use of these simpler forms in the later chapters, especially of (2.9). If we assume f to be lower semicontinuous for example, then

$$Df(x,d) \;=\; \lim \sup_{\substack{x_k \to x \\ f(x_k) \to f(x) \\ h_k \downarrow 0}} \inf_{d_k \to d} \frac{f(x_k + h_k d_k) - f(x_k)}{h_k} \,. \tag{2.7}$$

In the case of a continuous function f we can clearly simplify Equation (2.7) to

$$Df(x,d) \;=\; \lim \sup_{\substack{x_k \to x \\ h_k \downarrow 0}} \inf_{d_k \to d} \frac{f(x_k + h_k d_k) - f(x_k)}{h_k} \,. \tag{2.8}$$

The next step is to look at functions f that are Lipschitz continuous around x. To see how this affects (2.8), one has to go more into detail. The result is

$$Df(x,d) = \limsup_{\substack{x_k \to x \\ h_k \downarrow 0}} \frac{f(x_k + h_k d) - f(x_k)}{h_k} \,. \tag{2.9}$$

This important and last simplification turns out to be very useful in the field of numerical optimization. We can observe this in Chapters 5 and 6. A complete and rigorous treatment of how Equation (2.9) is justified can be found in [24, Thm.3]. From there we also cite the theorem that connects the Clarke derivative and the Clarke tangent cone [24, Thm.2]. It is a result comparable to Theorem 2.17.

Lemma 2.20.
Let $f : \mathbb{R}^n \to \mathbb{R}$ and $x \in \mathbb{R}^n$. Define $z(x) := \binom{x}{f(x)}$, then the following equation holds

$$epi\,(Df(x,\cdot)) = T\,(epi(f), z(x))\ . \tag{2.10}$$

This new formulation enables us to employ the concept of the Clarke derivative for set-valued mappings in Chapter 3. But we can benefit from it in various ways. First, we want to look again at an example.

Example 2.21.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be defined by $f(x) := 1 - e^{|x|}$. This by now familiar function was already introduced in Section 2.2. There we were able to graph $T\big(epi(f), z(x)\big)$ in Figure 2.5. From this picture we can immediately observe – using Lemma 2.20 – $Df(O,d) = |d|$. □

Second, we can combine Lemma 2.20 with the fact that $T\big(epi(f), z(x)\big)$ is a closed convex cone containing the origin (see Subsection 2.2.2). With our definition of sublinearity we directly find the next corollary.

Corollary 2.22.
Let $f : \mathbb{R}^n \to \mathbb{R}$ and $x \in \mathbb{R}^n$, then $Df(x,\cdot)$ is sublinear.

Finally, we get an idea of how to define further generalized derivatives. Assume we have a somehow defined cone of tangents for arbitrary points x on the boundary of a set S and denote it by $C(S,x)$. Then we can define for arbitrary functions $f : \mathbb{R}^n \to \mathbb{R}$ a derivative $Cf(x,\cdot) : \mathbb{R}^n \to \mathbb{R}$ by requesting of it

$$epi\big(Cf(x,\cdot)\big) = C\Big(epi(f), \tbinom{x}{f(x)}\Big)\ .$$

For example we could according to Subsection 2.2.3 define a *hyperderivative.* Instead of doing so, we concentrate on the Clarke derivative now. We exploit its properties, especially the sublinearity, in the following chapters.

Chapter 3

Set-Valued Analysis

We now focus on set-valued mappings and their analysis. Bearing in mind the applications of Chapters 5 and 6, we soon limit our point of view to special kinds of set-valued mappings in Section 3.1. These are generated from a single-valued function by surrounding each of its function values with an ε-interval. We then define for these set-valued mappings a modified version of the Clarke derivative that is consistent with the single-valued case. After that we show existence and uniqueness for it in Section 3.2. Then we define a minimizer for set-valued mappings as well as state and prove a necessary condition for it in Section 3.3. Finally, in Section 3.4 we transfer these results back to the single-valued case and discuss the relationship between the new and other classical necessary conditions.

3.1 Basic Definitions

In order to lay a foundation we have to clarify what a set-valued mapping is. Then we have to redefine some concepts from the single-valued framework in this setting.

Definition 3.1.
Let S and T be sets and 2^T denote the power-set of T, then a mapping $F : S \to 2^T$ is called a *set-valued mapping*. The set

$$F(S) := \bigcup_{x \in S} F(x)$$

is called the *image set of F*. $\square$

We concentrate on the special case $S = \mathbb{R}^n$ and $T = \mathbb{R}$. Furthermore we turn our attention to only one special class of set-valued mappings that turns out to be very useful for the numerical optimization of noisy functions. It is created by taking single-valued functions and surrounding them with a (small) ε-envelope.

Definition 3.2.
Let $f : \mathbb{R}^n \to \mathbb{R}$ and $f_-, f_+ : \mathbb{R}^n \to \mathbb{R}$ be functions such that $f_- \le f \le f_+$; then the set-valued mapping $F_f : \mathbb{R}^n \to 2^{\mathbb{R}}$ defined by

$$\forall\, x \in \mathbb{R}^n : \quad F_f(x) := \{y \in \mathbb{R} \,\big|\, f_-(x) \le y \le f_+(x)\} \tag{3.1}$$

is called an *envelope mapping around* f. □

Equivalently we could have defined F_f as

$$F_f(x) \;=\; \{y \in \mathbb{R} \,\big|\, f - \varepsilon_-(x) \le y \le f + \varepsilon_+(x)\} \tag{3.2}$$

for a pair of functions $\varepsilon_-, \varepsilon_+ : \mathbb{R}^n \to \mathbb{R}_0^+$. They are obviously related to f_- and f_+ by $\varepsilon_- = f - f_-$ and $\varepsilon_+ = f_+ - f$, respectively. This alternative formulation points out the ε-envelope character of F_f. We illustrate this idea by giving an example.

Example 3.3.
Let $f(x) := x^2$, $f_-(x) := \frac{1}{2}x^2$, $f_+(x) := 2x^2$ and F_f be given by (3.1). While the

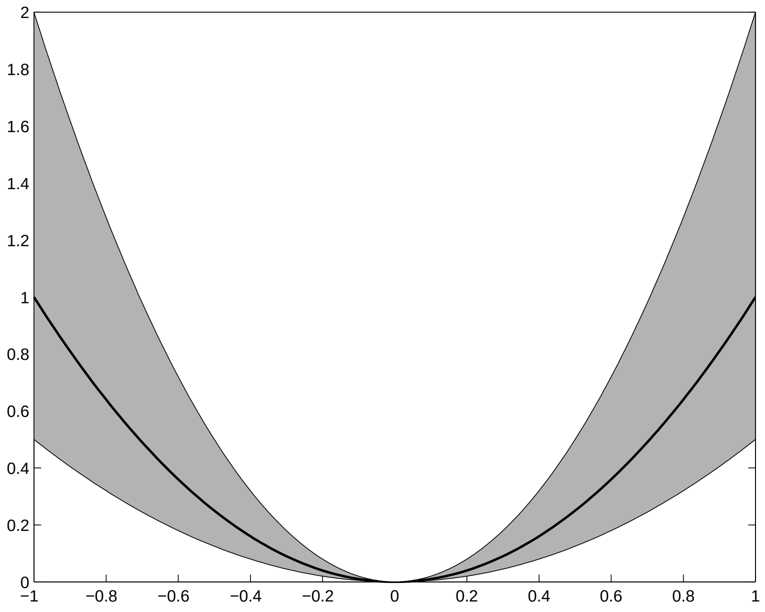

Figure 3.1: Example for an envelope mapping around $f(x) = x^2$

image set of the mapping F_f is shaded in grey, the graph of the underlying function f is plotted in black in Figure 3.1. We observe that $f(x) \in F_f(x)$ is always true. Especially, at $x = 0$ where the set-valued mapping F_f has only one single value, this means $F_f(0) = \{f(0)\}$. □

After having clarified the idea of a set-valued mapping, we want to generalize the definition of an epigraph given in Section 2.1. Note that this generalization reduces in the special case of a point-valued mapping to Definition 2.3 and thus is consistent with it.

Definition 3.4.
Let $F : \mathbb{R}^n \longrightarrow 2^{\mathbb{R}}$ be a set-valued mapping, then the set

$$epi(F) \;:=\; \left\{ \binom{x}{y} \in \mathbb{R}^n \times \mathbb{R} \;\middle|\; \exists\, \hat{y} \in F(x) \; s.t. \; y \geq \hat{y} \right\} \subset \mathbb{R}^{n+1}$$

is called the *epigraph of F*. □

We notice that the epigraph of the mapping F_f described by (3.1) equals the epigraph of the function f_- . Thus the corresponding Clarke tangent cones are also equal. This means if we define

$$\forall\, x \in \mathbb{R}^n : \quad z_-(x) := \binom{x}{f_-(x)} \in \mathbb{R}^{n+1} \;, \tag{3.3}$$

then

$$T\big(epi(F_f), z_-(x)\big) \;=\; T\big(epi(f_-), z_-(x)\big) \;. \tag{3.4}$$

The result of Lemma 2.20 now motivates the generalized definition of the Clarke derivative for set-valued mappings.

Definition 3.5.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x \in \mathbb{R}^n$ and F_f be an envelope mapping around f given by (3.1), then a single-valued function $DF_f(x,\cdot)$ that satisfies

$$epi\big(DF_f(x,\cdot)\big) = T\big(epi(F_f), z_-(x)\big) \tag{3.5}$$

is called the *Clarke derivative of F_f at x* . □

This definition brings up two questions. First, one has to ask if such a function exists. In the case of existence, we conclude

$$\begin{aligned} epi\big(DF_f(x,\cdot)\big) &\overset{(3.5)}{=} T\big(epi(F_f), z_-(x)\big) \\ &\overset{(3.4)}{=} T\big(epi(f_-), z_-(x)\big) \\ &\overset{(2.10)}{=} epi\big(Df_-(x,\cdot)\big) \;. \end{aligned} \tag{3.6}$$

From this observation the question of uniqueness arises. Can we conclude from Equation (3.6) that $DF_f(x,\cdot) = Df_-(x,\cdot)$? There is a positive answer to both questions. We can prove existence and uniqueness and we do this in the next section. Thus, (3.6) has some fruitful implications, which we reveal after the proofs and use in the subsequent chapters.

3.2 Existence and Uniqueness Issues

We certainly want to know under which circumstances we can take the existence of the Clarke derivative of an envelope mapping for granted. In the case of existence we additionally want to prove the uniqueness of this derivative. Let us start with the latter.

Theorem 3.6.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function, $F_f : \mathbb{R}^n \to 2^{\mathbb{R}}$ be an envelope mapping around f given by (3.1) and $x \in \mathbb{R}^n$. If the Clarke derivative $DF_f(x,\cdot)$ exists, then it is unique.

Proof:
Assume $DF_f^1(x,\cdot)$, $DF_f^2(x,\cdot)$ are Clarke derivatives and $DF_f^1(x,\cdot) \neq DF_f^2(x,\cdot)$, then

$$\exists\, d \in \mathbb{R}^n \text{ such that } DF_f^1(x,d) \neq DF_f^2(x,d) .$$

Definition 2.3 and Definition 3.5 then yield the following contradiction

$$\begin{aligned} T\big(epi(F_f), z_-(x)\big) &= epi\big(DF_f^1(x,\cdot)\big) \\ &\neq epi\big(DF_f^2(x,\cdot)\big) \\ &= T\big(epi(F_f), z_-(x)\big) . \end{aligned}$$

Thus, we have established uniqueness. ■

We want to remark that this uniqueness proof is also possible in a more general setting. We could define the Clarke derivative for a larger class of set-valued mappings. We would not even have to restrict the range of our mappings to $\mathbb{R}$. Any partially ordered real normed space is sufficient. Such a proof not for the Clarke derivative, but for a set-valued version of the contingent epiderivative can be found in [18].

Also the following existence statement could be generalized. But since Theorem 3.8 together with the assumptions made in Chapters 5 and 6 will guarantee the desired existence, we refrain from doing so. At first we show a lemma that will shorten our existence proof.

Lemma 3.7.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x \in \mathbb{R}^n$ and define $z(x) := \binom{x}{f(x)}$. Then for all $\binom{d}{y} \in \mathbb{R}^n \times \mathbb{R}$ with $\binom{d}{y} \in T\big(epi(f), z(x)\big)$, the following holds

$$\bar{y} > y \quad \Rightarrow \quad \binom{d}{\bar{y}} \in T\big(epi(f), z(x)\big) .$$

Proof:
Let $\binom{d}{y} \in T\big(epi(f), z(x)\big)$ and $\delta_1 > 0$ be arbitrary. Then by (2.3) we find

$$\exists\, \delta_2 > 0 \quad \forall \tbinom{x'}{f'} \in epi(f) \cap B_{\delta_2}\big(z(x)\big), h' \in (0, \delta_2) \quad \exists \tbinom{d'}{y'} \in B_{\delta_1}\big(\tbinom{d}{y}\big) :$$
$$\tbinom{x'}{f'} + h'\tbinom{d'}{y'} \in epi(f)\ .$$

This can be rewritten as

$$\exists\, \delta_2 > 0 \quad \forall \tbinom{x'}{f'} \in epi(f) \cap B_{\delta_2}\big(z(x)\big), h' \in (0, \delta_2) \quad \exists \tbinom{d'}{y'} \in B_{\delta_1}\big(\tbinom{d}{y}\big) :$$
$$f' + h'y' \geq f(x' + h'd')\ .$$

Now define for an arbitrary $\bar{y} > y$

$$\bar{y}' := y' + (\bar{y} - y) > y'$$

to conclude $\binom{d'}{\bar{y}'} \in B_{\delta_1}\big(\binom{d}{\bar{y}}\big)$ and

$$\forall \tbinom{x'}{f'} \in B_{\delta_2}(z(x)) \cap epi(f), h' \in (0, \delta_2) :$$
$$f' + h'\bar{y}' > f' + h'y' \geq f(x' + h'd')\ .$$

Thus by (2.3) we have shown that $\binom{d}{\bar{y}} \in T\big(epi(f), z(x)\big)$. ■

We are now prepared to prove the following existence statement.

Theorem 3.8.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x \in \mathbb{R}^n$ and F_f be an envelope mapping around f given by (3.1), then $DF_f(x, \cdot)$ exists and for $z_-(x)$ defined by (3.3) it is given by

$$\forall\, d \in \mathbb{R}^n : \quad DF_f(x, d) = \inf\big\{y \in \mathbb{R} \mid \tbinom{d}{y} \in T\big(epi(f_-), z_-(x)\big)\big\}\ . \tag{3.7}$$

Proof:
Equation (3.4) implies $T\big(epi(F_f), z_-(x)\big) = T\big(epi(f_-), z_-(x)\big)$. Thus, by Definition 3.5 we have to show for $DF_f(x, \cdot)$ defined by (3.7) and all $d \in \mathbb{R}^n$ that

$$E(d) \ := \ \{y \in \mathbb{R}^n \,|\, y \geq DF_f(x, d)\}$$

is equal to

$$T(d) \ := \ \big\{y \in \mathbb{R} \,|\, \tbinom{d}{y} \in T\big(epi(f_-), z_-(x)\big)\big\}\ .$$

For this purpose let $d \in \mathbb{R}^n$ be arbitrary. We have to distinguish the following three cases:

i) $\nexists\, y \in \mathbb{R}$ such that $\binom{d}{y} \in T\big(epi(f_-), z_-(x)\big)$

This directly implies

$$DF_f(x,d) = \infty$$

and thus

$$E(d) = \emptyset = T(d)\ .$$

ii) $\exists\, y_* \in \mathbb{R}$ such that $y_* = \inf T(d)$

By the definition of the infimum there exists $\{y_i\}_{i=1}^{\infty} \subset T(d)$ such that $y_i \downarrow y_*$. This implies

$$\tbinom{d}{y_i} \to \tbinom{d}{y_*} \ \text{ and } \ \forall\, i \in \mathbb{N}: \ \tbinom{d}{y_i} \in T\big(epi(f_-), z_-(x)\big)\ .$$

Since $T\big(epi(f_-), z_-(x)\big)$ is a closed set, we find $\binom{d}{y_*} \in T\big(epi(f_-), z_-(x)\big)$ and

$$DF_f(x,d) = y_* = \min T(d)\ .$$

By Lemma 3.7 we conclude

$$E(d) = T(d)\ .$$

iii) $\forall\, N \in \mathbb{N}\ \ \exists\, y \in T(d)$ such that $y < -N$

Again the definition of the infimum tells us

$$DF_f(x,d) = -\infty\ .$$

We use Lemma 3.7 once more to establish

$$E(d) = \mathbb{R} = T(d)\ .$$

Since these are all possible cases, we are finished with the proof. ■

Immediately following the definition of the Clarke derivative for an envelope mapping in Section 3.1, we already realized the connection to the Clarke derivative of f_-. Equation (3.6) and Theorem 3.8 together imply that the existence of $Df_-(x,\cdot)$ is equivalent to the existence of $DF_f(x,\cdot)$. Furthermore, if one of them exists, then Theorem 3.6 tells us that the two functions are equal. This observation enables us to transfer all results of Chapter 2 to the set-valued environment. We will make use of this immediately in the next section as well as in the oncoming chapters.

3.3 Clarke Derivative Based Necessary Condition

In order to make full use of the concept we just developed, we are interested in a necessary condition for optimization. In the corresponding theorem the Clarke derivative should play the central role normally taken by the classical directional derivative. Once we have derived this necessary condition, its usefulness has to be evaluated. We do this by comparing it with other necessary conditions that are based on strict differentiability, Fréchet differentiability, and the one-sided directional derivative, respectively.

Since we are concerned with set-valued mappings, we should first state how a minimizer of a set-valued mapping is defined in general. Afterwards we take a closer look at what this means for the special case of an envelope mapping.

Definition 3.9.
Let $F : \mathbb{R}^n \to 2^{\mathbb{R}}$ be a set-valued mapping, $x_* \in \mathbb{R}^n$ and $y_* \in F(x_*)$. Then the vector $z(x_*) := \binom{x_*}{y_*}$ is called a *global minimizer of* F, if

$$\forall\, y \in F(\mathbb{R}^n) : \quad y_* \le y \,. \tag{3.8}$$

In this case y_* is called a *global minimal element* of the image set $F(\mathbb{R}^n)$.

Let $B_\delta(x_*)$ denote the ball with radius δ around x_*, then the vector $z(x_*)$ is called a *local minimizer of* F, if

$$\exists\, \delta > 0 \ \text{ such that } \ \forall\, y \in F(B_\delta(x_*)) : \quad y_* \le y \tag{3.9}$$

and y_* is called a *local minimal element* of the image set $F(\mathbb{R}^n)$. □

For an envelope F_f defined by (3.1), this means that the global/local minimal element has to be a function value of f_- and even more it is clearly also a global/local minimum of this function.

Our aim is to use well-known facts about the single-valued function f_- to deduce facts about the corresponding envelope mapping – and at a later stage about the function f. To do so we need the results of the previous section. We discovered that if for an envelope mapping F_f defined by (3.1) the Clarke derivative exists, then

$$DF_f(x,d) = Df_-(x,d) \tag{3.10}$$

holds for all vectors d in $\mathbb{R}^n$. These are nearly all ingredients we need to prove the next theorem.

Theorem 3.10.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x_* \in \mathbb{R}^n$ and F_f an envelope mapping around f given by (3.1). If $z_-(x_*)$ defined by (3.3) is a local minimizer of F_f and the Clarke derivative $DF_f(x_*,\cdot)$ at x_* exists, then

$$\forall\, d \in \mathbb{R}^n : \qquad DF_f(x_*,d) \ge 0 \,. \tag{3.11}$$

Proof:
We prove the theorem indirectly by assuming

$$\exists\, d \in \mathbb{R}^n \text{ such that } y := DF_f(x_*, d) \overset{(3.10)}{=} Df_-(x_*, d) < 0\ . \tag{3.12}$$

Let $\delta > 0$ be arbitrary. With the remarks directly following Definition 3.9, it is sufficient to show that

$$\exists\, \bar{x} \in B_\delta(x_*) : \quad f_-(\bar{x}) < f_-(x_*)\ . \tag{3.13}$$

For this purpose we notice that (3.12) implies via (3.6)

$$\tbinom{d}{y} \in T(epi(f_-), z_-(x_*))\ .$$

This means by (2.3) that

$$\begin{aligned}&\forall\, \delta_1 > 0 \quad \exists\, \delta_2 > 0 \quad \forall\, z' \in epi(f_-) \cap B_{\delta_2}\big(z_-(x_*)\big)\ ,\ h' \in (0, \delta_2)\\ &\exists\, \tbinom{d'}{y'} \in B_{\delta_1}(\tbinom{d}{y}) : \quad z' + h'\tbinom{d'}{y'} \in epi(f_-)\ .\end{aligned}$$

If we rewrite $z' = \binom{x'}{f'}$, this means by the definition of an epigraph that

$$\begin{aligned}&\forall\, \delta_1 > 0 \quad \exists\, \delta_2 > 0 \quad \forall\, \tbinom{x'}{f'} \in epi(f_-) \cap B_{\delta_2}\big(z_-(x_*)\big)\ ,\ h' \in (0, \delta_2)\\ &\exists\, \tbinom{d'}{y'} \in B_{\delta_1}(\tbinom{d}{y}) : \quad f' + h'y' \geq f_-(x' + h'd')\ .\end{aligned} \tag{3.14}$$

We now choose $\delta_1 < |y|$ to guarantee

$$\forall\, \tbinom{d'}{y'} \in B_{\delta_1}(\tbinom{d}{y}) : \quad y' < 0\ . \tag{3.15}$$

Next we pick $\binom{x'}{f'} = z_-(x_*) \in epi(f_-) \cap B_{\delta_2}\big(z_-(x_*)\big)$ and find due to (3.14) that

$$\forall\, h' \in (0, \delta_2)\ \ \exists\, \tbinom{d'}{y'} \in B_{\delta_1}(\tbinom{d}{y}) : \ \ f_-(x_*) \overset{(3.15),(3.14)}{>} f_-(x_* + h'd')\ .$$

Finally we choose $h' \in (0, \delta_2)$ small enough such that

$$\bar{x} := x_* + h'd' \in B_\delta(x_*)\ .$$

Thus, we have shown (3.13) and our proof is complete. ∎

Note that although we stated this necessary condition with the Clarke derivative of the set-valued envelope mapping F_f, within the proof we only used the equivalent Clarke derivative of the single-valued function f_- . This technique shows up again in the succeeding chapters. It also tells us that there is a close relation between the envelope mapping F_f and its lower boundary function f_- . Especially we notice that Theorem 3.10 still holds if we replace the set-valued Clarke derivative by Df_- .

Of course this necessary condition is also true for the special case of point-valued mappings, i.e. single-valued functions. The functions only need to have a Clarke derivative. However, in order to use the Clarke derivative as a tool for convergence analysis we have to require the objective function to be Lipschitz continuous around the point x_*. This allows us to benefit from the simplified expression given in Equation (2.9). But first, we discuss what Theorem 3.10, in the special case of a single-valued function, can and cannot achieve in comparison with other necessary conditions.

3.4 Comparison of Necessary Conditions

We now want to examine how our new necessary condition performs when it is compared with others. To be precise: If in addition to Clarke derivative there exists another derivative, which of the corresponding necessary conditions is stronger, i.e. which of them does disqualify more candidate minimizers?

The first and strongest notion of differentiability we want to compare with the Clarke derivative is *strict differentiability.* A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be strictly differentiable at $x \in \mathbb{R}^n$, if

$$\exists\, g \in \mathbb{R}^n \;\; \forall\, d \in \mathbb{R}^n : \quad \lim_{\substack{x_k \to x \\ h_k \downarrow 0}} \frac{f(x_k + h_k d) - f(x_k)}{h_k} = d^T g \; . \tag{3.16}$$

The vector g is called the *gradient of f at x* and usually denoted by $\nabla f(x)$. We already used this symbol in Section 2.3 in connection with the Fréchet derivative. In fact it can be shown that if f is strictly differentiable, then f is Fréchet differentiable and the two derivatives are equal. Furthermore, in this case the Clarke derivative, which is

$$\begin{aligned} Df(x,d) &= \limsup_{\substack{x_k \to x \\ h_k \downarrow 0}} \frac{f(x_k + h_k d) - f(x_k)}{h_k} \\ &= \lim_{\substack{x_k \to x \\ h_k \downarrow 0}} \frac{f(x_k + h_k d) - f(x_k)}{h_k} \\ &= d^T \, \nabla f(x) \; , \end{aligned} \tag{3.17}$$

coincides with the other notions, too. Thus, if we assume strict differentiability for a necessary condition, then we can state it with either the gradient or the Clarke derivative. It makes no difference, because the concepts are equivalent.

On the other hand if we only require *Fréchet differentiability*, then the accompanying necessary condition is as follows.

Theorem 3.11.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x_* \in \mathbb{R}^n$ be a local minimizer of f. If the Fréchet derivative $f'(x_*, \cdot)$ of f at x_* exists, then

$$\forall\, d \in \mathbb{R}^n : \quad f'(x_*, d) = 0 \quad \textit{or equivalently} \quad \nabla f(x_*) = O \ .$$

By the definition of Fréchet and contingent epiderivative (compare Subsection 2.3.1) we find for $x, d \in \mathbb{R}^n$

$$f'(x,d) = \lim_{\substack{d_k \to d \\ h_k \downarrow 0}} \frac{f(x + h_k d_k) - f(x)}{h_k} = \liminf_{\substack{d_k \to d \\ h_k \downarrow 0}} \frac{f(x + h_k d_k) - f(x)}{h_k} = Ef(x,d) \ .$$

The results of Chapter 2 lead us to the conclusion

$$\begin{aligned} epi\big(Ef(x,\cdot)\big) &\overset{Lem.\ 2.17}{=} K\big(epi(f), z(x)\big) \\ &\overset{Tab.\ 2.1}{\supset} T\big(epi(f), z(x)\big) \\ &\overset{Lem.\ 2.20}{=} epi\big(Df(x,\cdot)\big) \ . \end{aligned}$$

But this implies for all $d \in \mathbb{R}^n$ that

$$Df(x,d) \geq Ef(x,d) = f'(x,d) \ .$$

Consequently Theorem 3.10 could be a weaker necessary condition. There might be an example where for all $d \in \mathbb{R}^n$ we have $Df(x,d) \geq 0$ but for at least one $d_0 \in \mathbb{R}^n$ we find $f'(x, d_0) < 0$. Thus, x would be a candidate minimizer of f after testing it with the Clarke derivative, but Theorem 3.11 would rule it out. There really are such examples. However, they are not easy to find.

Example 3.12.
Define the function, compare Figure 3.2

$$f(x) := \begin{cases} x^2 & x \in \mathbb{Q} \\ -2x - 1 & x \in \mathbb{R} \setminus \mathbb{Q} \end{cases} \ .$$

We first explain why f is Fréchet differentiable at $x = -1$ and $f'(-1, 1) = -2$. Note that both parts of f have the same slope of -2 at $x = -1$. Thus the difference quotient $\big(f(x + h_k d_k) - f(x)\big)/h_k$ converges to -2 as $h_k \downarrow 0$ and $d_k \to 1$, no matter whether we evaluate the function at rational or irrational points.

On the other hand we can show that $Df(-1, 1) = +\infty$. Since f is continuous at $x = -1$, we can use Equation (2.8) for that purpose. Further, we notice that for all $\{d_k\}_{k=1}^{\infty}$ such that $d_k \to 1$, there exist $\{x_k\}_{k=1}^{\infty}$, $\{h_k\}_{k=1}^{\infty}$ such that

$$x_k \to -1 \,, \quad h_k \downarrow 0 \,, \quad x_k \in \mathbb{R} \setminus \mathbb{Q} \,, \quad x_k + h_k d_k \in \mathbb{Q}$$

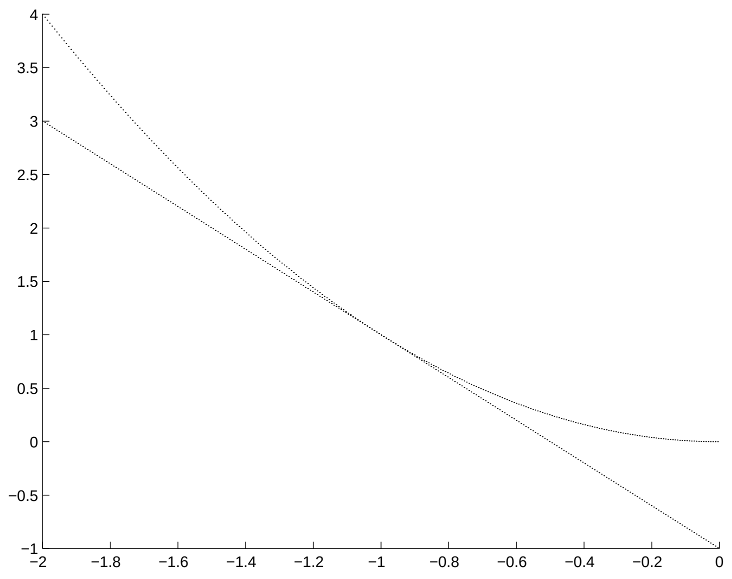

Figure 3.2: Graph of the function f, used in Example 3.12

and h_k is small enough to make the corresponding difference quotient monotonically increasing to infinity. □

This shows that the Clarke derivative in fact produces a weaker necessary condition. Another necessary condition involving the one-sided directional derivative does not require as much of the function f as Fréchet differentiability.

Theorem 3.13.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $x_* \in \mathbb{R}^n$ be a local minimizer of f. Then in the case of existence, the following holds

$$\forall\, d \in \mathbb{R}^n : \quad \lim_{h_k \downarrow 0} \frac{f(x_* + h_k d) - f(x_*)}{h_k} \quad \geq \quad 0\,. \tag{3.18}$$

We want to compare this necessary condition with Theorem 3.10 too. The existence of the directional derivative implies Lipschitz continuity around x_* for the function f. Thus, Equations (3.18) and (2.9) yield

$$\begin{aligned} Df(x_*, d) &= \limsup_{\substack{x_k \to x_* \\ h_k \downarrow 0}} \frac{f(x_k + h_k d) - f(x_k)}{h_k} \\ &\geq \limsup_{h_k \downarrow 0} \frac{f(x_* + h_k d) - f(x_*)}{h_k} \\ &\geq \lim_{h_k \downarrow 0} \frac{f(x_* + h_k d) - f(x_*)}{h_k}\,. \end{aligned}$$

Again the Clarke derivative seems to induce a weaker necessary condition. This time we can find a less pathological example to confirm this conjecture.

Example 3.14.
In Chapter 2 we used the function $g : \mathbb{R} \to \mathbb{R}$ defined by $g(x_1) = 1 - e^{|x_1|}$ to illustrate the different tangent cones. By rotating this function around the vertical axis, we get for $x = \binom{x_1}{x_2} \in \mathbb{R}^2$ the function $f(x) = 1 - e^{\|x\|}$.

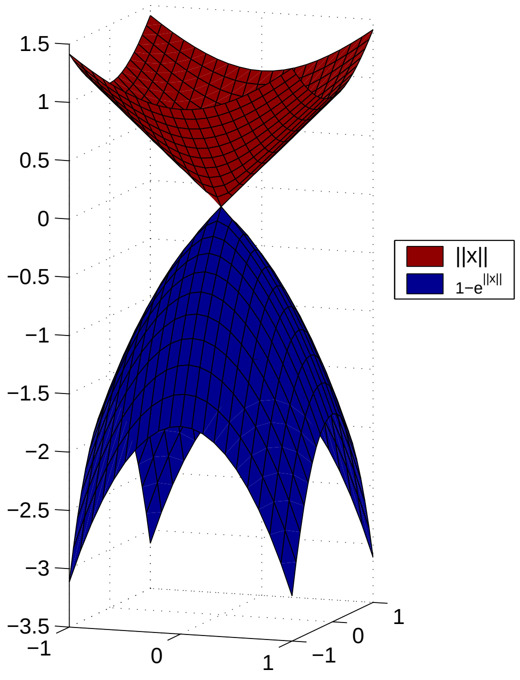

Figure 3.3: Graph (blue) and Clarke derivative at O (red) of $f(x) = 1 - e^{\|x\|}$

The Clarke derivative at the origin can be determined to be, compare Example 2.21,

$$\forall\, d \in \mathbb{R}^n : \quad Df(O,d) = \|d\| \geq 0 \,.$$

However, Figure 3.3 shows that f has no local minimum at the origin. In fact we find a local maximum there. This is confirmed by the directional derivative. Applying l'Hospital's rule we find

$$\forall\, d \in \mathbb{R}^n : \quad \lim_{h_k \downarrow 0} \frac{f(h_k d) - f(O)}{h_k} \;=\; -\|d\| \;\leq\; 0 \,.$$

□

These are so far the bad traits of the Clarke derivative. We have seen that in some cases the Clarke derivative is not as helpful as the classical notions of differentiation. However, this trade-off has to be expected, if one wants to take a more general approach. The new non-smooth convergence results, which we prove in the Chapters 5 and 6, compensate for this imperfection.

Chapter 4

Numerical Concepts

We now present the theoretical background from numerical analysis and linear algebra that is necessary for the following chapters. First, we introduce the idea of a positive spanning set in Section 4.1. Then we discuss various concepts of finite-difference approximations for the gradient of a function in Section 4.2. Namely these are the forward, backward and central simplex gradient. Finally, in Section 4.3 we present a particular line search procedure involving the steepest descent direction.

All these concepts help to understand the optimization routines we present in the Chapters 5 and 6. The Implicit Filtering algorithm, which is discussed in Chapter 5, is based on the ideas of finite-difference approximations and line search methods. The properties of a positive spanning set are important for the analysis of the Pattern Search methods of Chapter 6 as well as for Implicit Filtering. In combination with the sublinearity of the Clarke derivative, they are crucial for our 'derivative-free' convergence results.

4.1 Positive Spanning Sets

Positive spanning sets are related to the familiar idea of a general spanning or generating set. We remember that a generating of $\mathbb{R}^n$ set needs to consist of at least n vectors that are linearly independent. If we denote such a set by $\{v_i\}_{i=1}^m$ $(m \geq n)$, we know that

$$\forall\, x \in \mathbb{R}^n \;\exists\, \{\lambda_i\}_{i=1}^m \subset \mathbb{R} \;\text{ such that:}\quad x = \sum_{i=1}^m \lambda_i v_i \;.$$

Note that the coefficients $\{\lambda_i\}_{i=1}^m$ need not be unique. This is only the case, if $m = n$ and the vectors $\{v_i\}_{i=1}^m$ form a basis of $\mathbb{R}^n$. With this is mind we can understand the definition of a positive spanning set.

Definition 4.1.
A set $\{p_i\}_{i=1}^m \subset \mathbb{R}^n$ is called a *positive spanning set* of $\mathbb{R}^n$, if

$$\forall\, x \in \mathbb{R}^n\ \exists\ \{\lambda_i\}_{i=1}^m \subset \mathbb{R}_0^+ \ \text{ such that:} \quad x = \sum_{i=1}^m \lambda_i p_i\ .$$

□

This means that $\{p_i\}_{i=1}^m \subset \mathbb{R}^n$ is a special type of spanning set, i.e. the entire $\mathbb{R}^n$ is spanned by nonnegative linear combinations of this set. Obviously, a positive spanning set has to consist of at least $n+1$ vectors. Note that if we have a generating set $\{v_i\}_{i=1}^m$, then $\{v_i, -v_i\}_{i=1}^m$ is automatically a positive spanning set. We also want to mention that the coefficients $\{\lambda_i\}_{i=1}^m$ are never unique, because a positive spanning set is always linearly dependent. But this fact does not cause any problems for the later applications.

Positive spanning sets are important for us, because they allow to generalize inequalities for sublinear functions like the Clarke derivative. If $f : \mathbb{R}^n \to \mathbb{R}$ is a sublinear function and $\{p_i\}_{i=1}^m \subset \mathbb{R}^n$ is a positive spanning set such that

$$\forall\, i \in \{1, \ldots, m\} : \quad f(p_i) \geq 0\ ,$$

then by equation (2.1) we find

$$\forall\, x \in \mathbb{R}^n : \quad f(x) = f(\sum_{i=1}^m \lambda_i p_i) \geq \sum_{i=1}^m \lambda_i f(p_i) \geq 0\ .$$

We make use of this fact quite frequently. Why it is so useful becomes clearer in the subsequent chapters. It especially helps to show that the necessary condition of Theorem 3.10 is satisfied by a limit point x_*. The general idea for the proofs is that an algorithm provides a sequence of trial points and an associated positive spanning set. Assuming that a limit point x_* of the sequence exists, we first prove Equation (3.11) for the positive spanning set. Then the properties of sublinear functions and positive spanning sets enable us to deduce that x_* is a critical point.

4.2 Finite-Difference Gradients

In the next chapters we will analyze two special optimization algorithms. They have in common thatthey do not evaluate the exact gradient of the objective function. However, the Implicit Filtering method presented in Chapter 5 uses finite-difference approximations of the gradient. For the convergence analysis it is important to know how exact these approximations are. Thus, we now introduce this concept and exploit its accuracy. Most of the time we follow Kelley [19], who gives a good introduction to this matter. Before we attempt to define gradient approximations, we need to clarify some basic notions. We start with the convex hull of a set in $\mathbb{R}^n$.

Definition 4.2.
Let $S \subset \mathbb{R}^n$, then the set

$$conv(S) := \bigcap_{\substack{T \supset S \\ T \text{ convex}}} T$$

is called the *convex hull of S*. □

We need the convex hull to define a simplex in $\mathbb{R}^n$.

Definition 4.3.
Let $\{x_i\}_{i=1}^{n+1} \subset \mathbb{R}^n$, then the set

$$S = S(x_1, \dots, x_{n+1}) := conv(\{x_i\}_{i=1}^{n+1}) \subset \mathbb{R}^n$$

is called the *simplex with vertices* $x_1, \dots, x_{n+1}$ and

$$V = V(S) = (v_1, \dots, v_n) := (x_2 - x_1, \dots, x_{n+1} - x_1) \in \mathbb{R}^{n,n}$$

is called the *matrix of simplex directions*. The v_i $(i = 1, \dots, n)$ are the *simplex directions*. The simplex S is said to be *nonsingular*, if V is nonsingular. Furthermore, we define

$$\begin{array}{rcll} diam(S) & := & \max\limits_{1 \le i,j \le n+1} \|x_i - x_j\| & \text{as the } \textit{diameter of } S, \\ \sigma_+(S) & := & \max\limits_{1 \le i \le n} \|v_i\| & \text{as the } \textit{oriented length of } S, \\ \kappa(V) & := & \|V\|\|V^{-1}\| = \|V\|\|V^{-T}\| & \text{as the } \textit{simplex condition} \text{ and} \\ R = R(S) & := & conv(\{x_1; -x_2, \dots, -x_{n+1}\}) & \text{as the } \textit{reflected simplex.} \end{array}$$

□

After all these definitions some remarks are appropriate:

- Note that a simplex is singular, if and only if it has an n-dimensional volume of zero. As an example we present the two cases in $\mathbb{R}^2$ in the figure below.

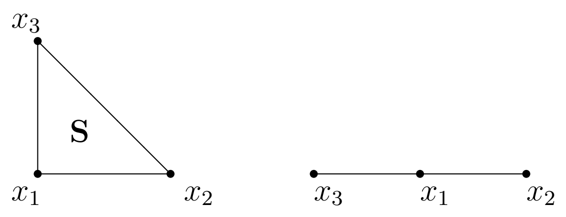

Figure 4.1: Nonsingular (left) and singular (right) simplex in $\mathbb{R}^2$

- The diameter is related to the oriented length by the following inequality

$$\sigma_+(S) \leq diam(S) \leq 2\sigma_+(S)$$

What follows is an example that should help to get an idea of this basic concept.

Example 4.4.
Let $h > 0$ and consider the simplex $S = conv(\{O, he_1, \ldots, he_n\})$, where the e_i denote the canonical unit vectors of $\mathbb{R}^n$. It is clearly nonsingular and $\sigma_+(S) = h$. Moreover if we denote the identity matrix in $\mathbb{R}^{n,n}$ by I_n, then

$$\kappa\big(V(S)\big) = \|V\|\|V^{-1}\| = h\|I_n\|\, h^{-1}\|I_n\| = \|I_n\|^2 = 1 \; .$$

□

Now we are well equipped to define various approximations for the gradient of a smooth function.

Definition 4.5.
Let $f : \mathbb{R}^n \to \mathbb{R}$, $\{x_i\}_{i=1}^{n+1} \subset \mathbb{R}^n$ and $S = conv(\{x_i\}_{i=1}^{n+1})$, then the *vector of function differences* $\Delta(f, S)$ *of* f *on* S is defined by

$$\Delta(f,S) := \begin{pmatrix} f(x_2) & - & f(x_1) \\ & \vdots & \\ f(x_{n+1}) & - & f(x_1) \end{pmatrix} \in \mathbb{R}^n \; .$$

Furthermore define

$$\begin{aligned} D^+(f,S) &:= V^{-T}\Delta(f,S) \quad , \\ D^-(f,S) &:= D^+(f,R) \quad \text{and} \\ D^c(f,S) &:= \frac{1}{2}\big(D^+(f,S) + D^-(f,S)\big) = D^c(f,R) \; . \end{aligned}$$

We call $D^+(f,S)$ the *forward finite-difference approximation of* $\nabla f(x_1)$ *on the simplex* S or, shorter, the *forward simplex gradient*. Likewise, $D^-(f,S)$ and $D^c(f,S)$ are called *backward* and *central simplex gradient*, respectively. □

At this point we want to draw the reader's attention to two facts. First, we always think of the gradient as a column vector. Second, the forward, backward and central simplex gradient are not the exact gradient or some other notion of a gradient as these short forms of their names might suggest. They are only attempts to approximate it. Again some examples shall illustrate the last definition.

Example 4.6.
If $n = 1$, we observe for $h > 0$, $x_1 \in \mathbb{R}$ and $x_2 = x_1 + h$ that our definitions coincide with the well-known one-dimensional finite-difference approximations for the first derivative.

$$\begin{aligned}
D^+(f, conv(\{x_1, x_2\})) &= \frac{f(x_1 + h) - f(x_1)}{h} \\
D^-(f, conv(\{x_1, x_2\})) &= \frac{f(x_1 - h) - f(x_1)}{-h} = \frac{f(x_1) - f(x_1 - h)}{h} \\
D^c(f, conv(\{x_1, x_2\})) &= \frac{1}{2}\left(\frac{f(x_1 + h) - f(x_1)}{h} + \frac{f(x_1) - f(x_1 - h)}{h}\right) \\
&= \frac{f(x_1 + h) - f(x_1 - h)}{2h}
\end{aligned}$$

□

Example 4.7.
To get a first intuition why for example $D^+(f, S)$ is a good approximation for the gradient, we set for $h > 0$: $x_1 = O$, $x_i = h\, e_{i-1}$ $(i = 2, \dots, n+1)$ and $S = conv(\{x_i\}_{i=1}^{n+1})$. From this we get, compare also Example 4.4,

$$V(S) = hI_n \quad , \quad V^{-T} = h^{-1}I_n \quad , \quad \kappa(V) = 1 \quad \text{and}$$

$$\Delta(f, S) = \begin{pmatrix} f(he_1) & - & f(O) \\ & \vdots & \\ f(he_n) & - & f(O) \end{pmatrix} .$$

Finally the forward simplex gradient is

$$D^+(f, S) = V^{-T}\Delta(f, S) = \begin{pmatrix} \frac{1}{h}\big(f(he_1) - f(O)\big) \\ \vdots \\ \frac{1}{h}\big(f(he_n) - f(O)\big) \end{pmatrix} .$$

Thus, $D^+(f, S)$ is a componentwise approximation of $\nabla f(O)$. □

As a next step we examine the accuracy of the forward and backward simplex gradient.

Lemma 4.8.
Let $S = conv(\{x_i\}_{i=1}^{n+1}) \subset \mathbb{R}^n$ be a nonsingular simplex and $f : \mathbb{R}^n \to \mathbb{R}$. If ∇f is Lipschitz continuous in a neighborhood of $S \cup R(S)$, then there exists a constant $K > 0$ such that

$$\begin{aligned}
\|\nabla f(x_1) - D^+(f, S)\| &\le K\,\kappa\big(V(S)\big)\,\sigma_+(S) \qquad \textit{and} & (4.1)\\
\|\nabla f(x_1) - D^-(f, S)\| &\le K\,\kappa\big(V(S)\big)\,\sigma_+(S) . & (4.2)
\end{aligned}$$

Proof:
We prove (4.1) first, then (4.2) is a direct consequence of Definition 4.5.

i) the forward simplex gradient

Let L denote the Lipschitz constant of ∇f, then we find for all $i \in \{1, \dots, n\}$

$$|f(x_{i+1}) - f(x_1) - v_i^T \nabla f(x_1)| \overset{Taylor}{\leq} L\|v_i\|^2 \leq L\big(\sigma_+(S)\big)^2 ,$$

where v_i are the columns of $V = V(S)$. This implies by the definition of the Euclidean norm

$$\|\Delta(f,S) - V^T \nabla f(x_1)\| \;\leq\; \sqrt{n} L \left(\sigma_+(S)\right)^2 . \tag{4.3}$$

The last step is now

$$\begin{aligned} \|D^+(f,S) - \nabla f(x_1)\| &= \|V^{-T}\Delta(f,S) - V^{-T}V^T\nabla f(x_1)\| \\ &\leq \|V^{-T}\|\|\Delta(f,S) - V^T\nabla f(x_1)\| \\ &\overset{(4.3)}{\leq} L\sqrt{n}\,\|V^{-1}\|\left(\sigma_+(S)\right)^2 \\ &\overset{\sigma_+(S)\geq\|V\|}{\leq} L\sqrt{n}\,\kappa\big(V(S)\big)\,\sigma_+(S) , \end{aligned}$$

which yields (4.1) by setting $K = L\sqrt{n}$.

ii) the backward simplex gradient

Since $D^-(f,S) = D^+(f,R)$ and $\|V(R)\| = \|-V(S)\| = \|V(S)\|$, inequality (4.2) follows immediately from the results of *i*).

■

A similar result can be shown for the central simplex gradient by using Taylor's formula for twice continuously differentiable functions. We only state the lemma and omit the proof, which can be found in [19, Lemma 6.2.5].

Lemma 4.9.
Let $S = conv(\{x_i\}_{i=1}^{n+1}) \subset \mathbb{R}^n$ be a nonsingular simplex and $f : \mathbb{R}^n \to \mathbb{R}$. If $\nabla^2 f$ is Lipschitz continuous in a neighborhood of $S \cup R(S)$, then there exists a constant $K > 0$ such that

$$\|\nabla f(x_1) - D^c(f,S)\| \;\leq\; K\,\kappa\big(V(S)\big)\left(\sigma_+(S)\right)^2 .$$

However, these results are not enough to make use of the approximations in the convergence analysis of Implicit Filtering in Chapter 5. This is based on the fact that the algorithm is designed for the optimization of noisy functions $f_c : \mathbb{R}^n \to \mathbb{R}$ of the form

$$f_c \;=\; f_s + \varphi\,. \tag{4.4}$$

Here f_c represents a function, whose values can be calculated. But actually we want to optimize the function f_s, which is supposed to be sufficiently smooth. The remaining φ can be thought of as some bounded function that resembles the existing error or noise. See Chapter 5 for more details. The next theorems provide suitable results for such a situation.

Theorem 4.10.
Let $S = conv(\{x_i\}_{i=1}^{n+1}) \subset \mathbb{R}^n$ be a nonsingular simplex and $f_c : \mathbb{R}^n \to \mathbb{R}$ be given by (4.4). If ∇f_s is Lipschitz continuous on a neighborhood of $U := S \cup R(S)$, then there exists a constant $K > 0$ such that

$$\|\nabla f_s(x_1) - D^+(f_c, S)\| \;\le\; K\,\kappa\big(V(S)\big)\left(\sigma_+(S) + \frac{\|\varphi(x)\|_U}{\sigma_+(S)}\right) \quad \textit{and} \tag{4.5}$$

$$\|\nabla f_s(x_1) - D^-(f_c, S)\| \;\le\; K\,\kappa\big(V(S)\big)\left(\sigma_+(S) + \frac{\|\varphi(x)\|_U}{\sigma_+(S)}\right). \tag{4.6}$$

Proof:
We only show (4.5), because then Definition 4.5 again implies immediately (4.6).

$$\begin{aligned}
\|\nabla f_s(x_1) - D^+(f_c, S)\| &= \|\nabla f_s(x_1) - D^+(f_s, S) + D^+(f_s, S) - D^+(f_c, S)\| \\
&\overset{(4.4)}{\le} \|\nabla f_s(x_1) - D^+(f_s, S)\| + \|D^+(\varphi, S)\| \\
&\overset{(4.1)}{\le} K_1\,\kappa\big(V(S)\big)\,\sigma_+(S) + \|V^{-1}\|\,\|\Delta(\varphi, S)\| \\
&\overset{\sigma_+(S)\le\|V\|}{\le} K_1\,\kappa\big(V(S)\big)\,\sigma_+(S) + \kappa\big(V(S)\big)\,\frac{2\sqrt{n}\,\|\varphi(x)\|_U}{\sigma_+(S)} \\
&\le K\,\kappa\big(V(S)\big)\left(\sigma_+(S) + \frac{\|\varphi(x)\|_U}{\sigma_+(S)}\right)
\end{aligned}$$

For the last inequality we have to choose $K \ge \max\{K_1, 2\sqrt{n}\}$. ■

There exists an analogous version for the central simplex gradient. Since for its proof we would have to use Lemma 4.9, the result is slightly different from (4.5) and (4.6).

Theorem 4.11.
Let $S = conv(\{x_i\}_{i=1}^{n+1}) \subset \mathbb{R}^n$ be a nonsingular simplex and $f_c : \mathbb{R}^n \to \mathbb{R}$ be given by (4.4). If $\nabla^2 f_s$ is Lipschitz continuous on a neighborhood of $U := S \cup R(S)$, then there exists a constant $K > 0$ such that

$$\|\nabla f_s(x_1) - D^c(f_c, S)\| \ \leq \ K\,\kappa\big(V(S)\big)\left(\big(\sigma_+(S)\big)^2 + \frac{\|\varphi(x)\|_U}{\sigma_+(S)}\right) . \tag{4.7}$$

Besides a better accuracy, the central simplex gradient has yet another advantage over the other approximations. Since none of them is equal to the exact gradient, they may fail to provide a descent direction like the gradient does; compare Section 4.3. But in various algorithms, like for example Implicit Filtering, this is what the approximations are used for. One wants at least to be warned if such a problem is encountered. Using the central simplex gradient, we can characterize one such situation and check if it occurs.

Definition 4.12.
Let $S = conv(\{x_i\}_{i=1}^{n+1}) \subset \mathbb{R}^n$ be a nonsingular simplex with simplex directions $\{v_i\}_{i=1}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ be given. If

$$\forall\, i \in \{1, \ldots, n\} : \quad f(x_1) < f(x_1 \pm v_i) \tag{4.8}$$

holds, then this situation is called a stencil failure. □

In this case none of the directions v_i $(i = 1, \ldots, n)$ is likely to provide a descent direction. But the following theorem shows that by using central finite-difference approximations we can conclude more than this.

Theorem 4.13.
Let $S = conv(\{x_i\}_{i=1}^{n+1}) \subset \mathbb{R}^n$ be a nonsingular simplex with a matrix of simplex directions $V = V(S)$ such that

$$\exists\, \mu \in (0,1) \ \forall\, x \in \mathbb{R}^n : \quad x^T V V^T x \geq \mu\, \big(\sigma_+(S)\big)^2\, \|x\|^2 . \tag{4.9}$$

Let $f_c : \mathbb{R}^n \to \mathbb{R}$ be given by (4.4). If ∇f_s is Lipschitz continuous on a neighborhood of $U := S \cup R(S)$ and (4.8) holds, then there exists a constant $K > 0$ such that

$$\|\nabla f_s(x_1)\| \ < \ K\,\frac{\kappa(V)}{\mu}\left(\sigma_+(S) + \frac{\sup_{x\in U} |\varphi(x)|}{\sigma_+(S)}\right) .$$

Proof:
By looking at (4.8) we find that each component of $\Delta(f_c, S)$ and $\Delta(f_c, R(S))$ is positive. Since $V = V(S) = -V(R(S)) =: -V(R)$ holds, we have

$$\begin{aligned} 0 \ &< \ \Delta(f_c, S)^T \Delta(f_c, R) \\ &= \ \big(V^T V^{-T} \Delta(f_c, S)\big)^T \ \big(V(R)^T V(R)^{-T} \Delta(f_c, R)\big) \\ &= \ -D^+(f_c, S)^T\ V V^T\ D^-(f_c, S) . \end{aligned}$$

Lemma 4.10 can now be used to find

$$0 \;<\; -(\nabla f_s(x_1) + E^+)^T \; VV^T \; (\nabla f_s(x_1) + E^-)\;,$$

where

$$\|E^\pm\| \le K_1 \, \kappa(V) \left(\sigma_+(S) + \frac{\|\varphi(x)\|_U}{\sigma_+(S)} \right) . \tag{4.10}$$

The former inequality can be rearranged to

$$\begin{aligned}\big(\nabla f_s(x_1)\big)^T \; VV^T \; \nabla f_s(x_1) \;&<\; -E^+VV^T\nabla f_s(x_1) - E^+VV^TE^- \\ &\quad\; -\nabla f_s(x_1)VV^TE^- \\ &\le\; \|\nabla f_s(x_1)\| \; \|V\|^2 \; (\|E^+\| + \|E^-\|) \\ &\quad\; +\|V\|^2 \; \|E^+\| \; \|E^-\| \; .\end{aligned}$$

Now the well-known fact that $\|V\| \le \sqrt{n}\,\sigma_+(S)$ shows us together with (4.9) applied for $x = \nabla f_s(x_1)$

$$\begin{aligned}\mu \, \big(\sigma_+(S)\big)^2 \; \|\nabla f_s(x_1)\|^2 \;&<\; n\big(\sigma_+(S)\big)^2 \; (\|E^+\| + \|E^-\|) \; \|\nabla f_s(x_1)\| \\ &\quad\; +n\big(\sigma_+(S)\big)^2 \; \|E^+\| \; \|E^-\| \; .\end{aligned}$$

Since the quantities $\big(\sigma_+(S)\big)^2$ can be omitted, we can use (4.10) to get

$$\begin{aligned}\mu \;\; \|\nabla f_s(x_1)\|^2 \;&<\; 2n \; K_1 \, \kappa(V) \left(\sigma_+(S) + \frac{\|\varphi(x)\|_U}{\sigma_+(S)} \right) \|\nabla f_s(x_1)\| \\ &\quad\; +n \left(K_1 \; \kappa(V) \; \left(\sigma_+(S) + \frac{\|\varphi(x)\|_U}{\sigma_+(S)} \right) \right)^2 \; .\end{aligned}$$

Finally we can use the formulas for a perfect square and a little arithmetic to find

$$\|\nabla f_s(x_1)\| < \frac{3\,n}{\mu} \;\; K_1 \; \kappa(V) \; \left(\sigma_+(S) + \frac{\|\varphi(x)\|_U}{\sigma_+(S)} \right) .$$

If we choose $K > 3\,n\,K_1$, the proof is complete. ∎

This also concludes the section about finite-difference approximations and we now turn our attention to line search methods.

4.3 A Line Search Method

It is a well-known fact that the efficient Newton method only converges locally. Line search methods are a way to globalize convergence. We encounter a practical implementation of this procedure in Chapter 5. Before that, we want to discuss the theoretical issues. A very general approach to this matter can be found in the book of Dennis and Schnabel [11]. We just look at a particular example that is necessary to understand the Implicit Filtering algorithm in the next chapter. However, our reasoning follows mainly the ideas of [11, Ch. 6]. The problem we are concerned with in this section is

$$\min_{x \in \mathbb{R}^n} f(x) \ .$$

We are looking for an iterative method that, given any initial $x_0 \in \mathbb{R}^n$, converges to a minimum of f. However, in general it is impossible to find such a method. Our expectations have to be lowered as we will see in this section.

Since we are looking for a minimizer, it seems appropriate to require for any two subsequent iterates x_k, x_{k+1} that

$$f(x_{k+1}) \leq f(x_k) \ . \tag{4.11}$$

In order to find x_{k+1} knowing only x_k, we search on a line segment containing x_k, along which f is supposed to decrease locally. The name *line search method* arose from this idea. We can enforce local decrease following a so-called descent direction of f.

Definition 4.14.
Let $x, p \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable around x. If

$$p^T \nabla f(x) < 0 \ ,$$

then p is called a *descent direction of f at x*. □

It is an easy application of Taylor's formula to justify the name descent direction. If there exists a descent direction p_1, then there are infinitely many of them, because $p^T \nabla f(x)$ is a continuous function of p. There also exist several different strategies for the line search along one of those directions. To make the transfer to the special algorithm in Chapter 5 easier, we focus here explicitly on the so-called steepest descent direction $p = -\nabla f(x_k)$. Thus, our new iterate will be of the form $x_{k+1} = x_k - \lambda \nabla f(x_k)$ for some step size $\lambda > 0$. The name steepest descent originates from the fact that

$$\frac{-\nabla f(x_k)^T}{\|\nabla f(x_k)\|} \nabla f(x_k) = \min_{\substack{p \in \mathbb{R}^n \\ \|p\|=1}} p^T \nabla f(x_k) \ .$$

To pick in addition a suitable line search procedure, we first mention the two problems that can occur:

1. The decrease we find may be too small.
2. The step size we choose may be too small.

Examples for both failures can be found in [11]. We take care of the latter by testing a sufficiently large stepsize λ first. If we do not find the desired decrease, we reduce λ successively. Then the first step size that yields a sufficient decrease is large enough. The last question that is left is: "What do we mean by *sufficient decrease*?" We do not accept just any x_{k+1} satisfying (4.11). Instead we require for some $\alpha \in (0,1)$:

$$f(x_k - \lambda \nabla f(x_k)) - f(x_k) < -\alpha \, \lambda \, \nabla f(x_k)^T \nabla f(x_k) \; . \tag{4.12}$$

This so-called sufficient decrease condition can be interpreted as follows. The right hand side of (4.12) is nothing else than a fraction of the decrease of the linearization of f in the steepest descent direction. We want the absolute value of the actually observed decrease $f\big(x_k - \lambda \nabla f(x_k)\big) - f(x_k)$ to be greater than that. Figure 4.2 illustrates this idea. While for λ_1 (4.12) is not satisfied, reducing the step size to

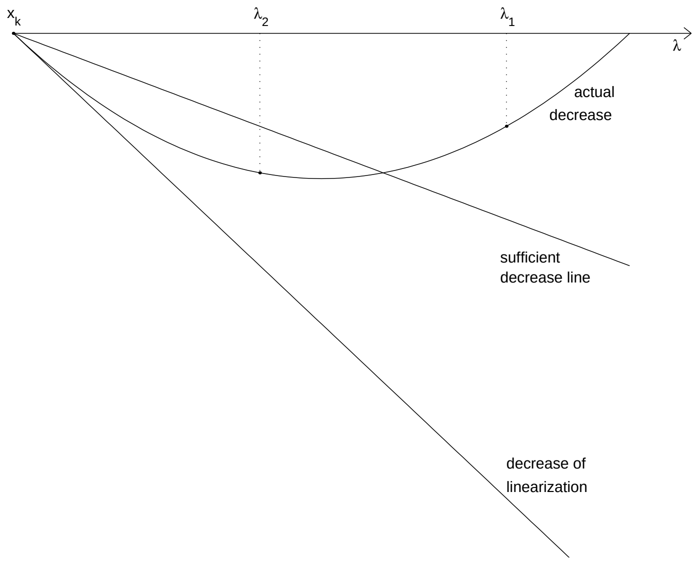

Figure 4.2: Visualization of the sufficient decrease condition

λ_2 yields sufficient decrease. The following lemma assures that we can always find a $\lambda > 0$ such that (4.12) is satisfied. Thus, our strategy is well-defined and can be applied to practical problems.

Lemma 4.15.
Let $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable around x, then for any $\alpha \in (0,1)$ there exists a step size $\lambda > 0$ that satisfies (4.12).

The proof is again no more than a corollary of Taylor's theorem. Now that we know about all the details, it is time to present the whole algorithm.

Algorithm: Line Search

```
Set $\alpha, \beta \in (0,1)$
Set $\tau$, the termination parameter
Set $\lambda$, the initial step size
Set $k = 0$

Initialize a starting point $x_0$
Compute the gradient $\nabla f(x_0)$

While $\|\nabla f(x_k)\| < \tau$

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

       If $f(x_k - \lambda\beta^l \nabla f(x_k)) - f(x_k) < -\alpha\ \lambda\ \beta^l\ \nabla f(x_k)^T \nabla f(x_k)$:  Exit For

    End For

    Update $x_{k+1} = x_k - \lambda\beta^l \nabla f(x_k)$

    Compute the gradient $\nabla f(x_{k+1})$

    Set $k = k + 1$

End While
```

For this algorithm we can prove the following convergence result. Most of the ideas for the proof are the same as in [11]. However, we present the whole derivation since it contains some concepts that we need again in the next section.

Theorem 4.16.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be bounded below and Lipschitz continuously differentiable on $\mathbb{R}^n$ with Lipschitz constant L. If $\{x_k\}_{k=0}^{\infty}$ is the sequence of iterates produced by the line search algorithm for f, then there exists an $f_* \in \mathbb{R}$ such that

i) $\lim_{k\to\infty} f(x_k) = f_*$ *and*

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

Thus every limit point of $\{x_k\}_{k=0}^{\infty}$ is a critical point of f.

Proof:
We prove the two parts of the theorem separately.

i) $\{f(x_k)\}_{k=0}^{\infty}$ is monotonically decreasing because of (4.12). By the boundedness of f we already find $\exists\, f_* := \lim_{k\to\infty} f(x_k)$.

ii) We have to distinguish two cases. On the one hand the step size in iteration k, call it λ_k, could be λ; on the other hand it could be $\lambda\beta^l$ ($l \in \mathbb{N}$) as well.

Assume first that $\lambda_k = \lambda\beta^l$, then

$$\begin{aligned}
-\alpha\frac{\lambda_k}{\beta}\|\nabla f(x_k)\|^2 &= -\alpha\frac{\lambda_k}{\beta}\nabla f(x_k)^T\nabla f(x_k) \\
&\overset{(4.12)}{<} f\big(x_k - \frac{\lambda_k}{\beta}\nabla f(x_k)\big) - f(x_k) \\
&\overset{F.L.o.C.}{=} -\int_0^1 \frac{\lambda_k}{\beta}\nabla f(x_k)^T\nabla f\big(x_k - t\frac{\lambda_k}{\beta}\nabla f(x_k)\big)dt \\
&\overset{\pm 0}{=} -\frac{\lambda_k}{\beta}\int_0^1 \nabla f(x_k)^T\big[\nabla f\big(x_k - t\frac{\lambda_k}{\beta}\nabla f(x_k)\big) - \nabla f(x_k)\big]dt \\
&\quad -\frac{\lambda_k}{\beta}\nabla f(x_k)^T\nabla f(x_k) \\
&\overset{C.S.I.}{\leq} \frac{\lambda_k}{\beta}\int_0^1 \|\nabla f(x_k)\|\,\|\nabla f\big(x_k - t\frac{\lambda_k}{\beta}\nabla f(x_k)\big) - \nabla f(x_k)\|dt \\
&\quad -\frac{\lambda_k}{\beta}\|\nabla f(x_k)\|^2 \\
&\overset{Lip.}{\leq} \frac{\lambda_k}{\beta}\|\nabla f(x_k)\|\int_0^1 t\,L\,\|\frac{\lambda_k}{\beta}\nabla f(x_k)\|dt - \frac{\lambda_k}{\beta}\|\nabla f(x_k)\|^2 \\
&= \frac{L}{2}\Big(\frac{\lambda_k}{\beta}\|\nabla f(x_k)\|\Big)^2 - \frac{\lambda_k}{\beta}\|\nabla f(x_k)\|^2\ .
\end{aligned}$$

If we now multiply both ends with $\frac{2\beta^2}{L\,\lambda_k\,\|\nabla f(x_k)\|^2}$ and rearrange the inequality, we find

$$\lambda_k \ > \ \frac{2(1-\alpha)\beta}{L}\ .$$

If we consider in addition the second case $\lambda_k = \lambda$, we find in general

$$\lambda_k \geq \min\Big\{\lambda, \frac{2(1-\alpha)\beta}{L}\Big\}\ . \tag{4.13}$$

Now the result from $i)$ yields

$$0 > -\alpha\,\frac{\lambda_k}{\beta}\,\|\nabla f(x_k)\|^2 \overset{(4.12)}{>} f(x_{k+1}) - f(x_k) \overset{k\to\infty}{\longrightarrow} 0\ .$$

Since (4.13) holds, we can finally conclude that

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

■

We want to end this chapter with a warning. *The sequence of iterates does **not** necessarily converge.* We only know that if it does, then the limit point is a critical point of f. One has to pay even more attention in the analysis of the Implicit Filtering method, which is presented next. We learn in Section 5.2 that there is no guarantee for the success of the line search within this method. We have to turn this into an assumption and hope that the algorithm performs well.

Chapter 5

Implicit Filtering

The previous chapters equipped us with the necessary tools in optimization. Now the time has come to explore possible applications. The analysis of numerical algorithms for solving minimization problems for complex systems is such a field. The first algorithm we want to examine is called Implicit Filtering. It was originally formulated in [6], [14], [15] and has been developed for the optimization of a particular type of function f_c. The function can be split up into a *smooth* part $f_s \in C^1(\mathbb{R}^n)$ and a bounded non-smooth part $\varphi : \mathbb{R}^n \to \mathbb{R}$. Their relation is modelled by the following equation

$$\forall\, x \in \mathbb{R}^n : \qquad f_c(x) = f_s(x) + \varphi(x)\ . \tag{5.1}$$

The idea is that f_s describes the exact behavior of the quantity we want to optimize. However, due to some error – also referred to as noise – we can only *compute* function values of the perturbed function f_c. An example for such a situation is the remediation of ground water contaminations presented in [5]. The bounded function φ describes the immanent noise, which may be caused by various reasons. Namely they can be

- inaccurate models of the real situation,
- an insufficient amount of empirical data and/or
- inexact numerical computations.

This allows f_c to have several local minima, even if f_s is a convex function. Thus, conventional algorithms for smooth optimization are not appropriate. In addition it is presumed that φ may not return the same value when called twice with the same argument (see [19]). This complicates the situation even more. It also suggests the use of set-valued optimization techniques as described in Chapter 3. We justify this approach in Section 5.3.

Implicit Filtering is an attempt to handle the mentioned problems. At first we describe the basic algorithm in Section 5.1. Then we present the known convergence results in Section 5.2. They are based on assuming Fréchet differentiability for f_s. We discuss this and various other assumptions. Finally, we show in Section 5.3 that one of them can be dropped. The Fréchet differentiability becomes superfluous, when we use the set-valued approach of the previous chapters.

5.1 Description of the Algorithm

The basic unconstrained form of Implicit Filtering can be seen as a repeated call of the steepest descent line search described in Section 4.3. It uses one of the simplex gradients we discussed in Section 4.2 to calculate an approximation for the steepest descent direction. For every iterate we use the simplex directions $he_1, \dots, he_n$ and/or $-he_1, \dots, -he_n$ (depending on the implemented approximation). The so-called difference increments or scales h are positive and decrease while the algorithm progresses. Their variation is an attempt to filter the high frequency noise, introduced by the function φ, at different levels (compare [13]). This idea motivates the name Implicit Filtering.

What follows is the schematic description of a prototype algorithm. Since the finite-difference approximations only depend on the objective function f, the current iterate x and the used scale h, we denote them by $D_h f(x)$. Note that we also test for stencil failure, if the used simplex gradient allows to do so, compare Section 4.2.

Algorithm: Basic Implicit Filtering

```
Set MINH, MAXH as limits for the scales h
Set MAXIT for the admissible number of line search iterations
Set MAXCUT for the admissible number of line search backtracking steps
Set $\alpha$, $\beta$, $\lambda$ as additional line search parameters
Set $\tau$ as termination tolerance for $\|D_h f(x)\| \leq \tau \cdot h$

Initialize the starting point x
Initialize the first scale h=MAXH

While $h \geq$ MINH do

   For1 $m = 1, \dots,$ MAXIT

      Compute the finite difference approximation $D_h f(x)$

      If possible test for stencil failure:

         If stencil failure occurs:  Exit For1

      If $\|D_h f(x)\| \leq \tau \cdot h$ is satisfied:  Exit For1

      For2 $l = 0, \dots,$ MAXCUT

         Set $\lambda_l = \lambda\beta^l$

         If $f(x - \lambda_l D_h f(x)) - f(x) < -\alpha\lambda_l \|D_h f(x)\|^2$:
            Update $x = x - \lambda_l D_h f(x)$
            Exit For2
          Else:  If $l = $ MAXCUT:  Exit For1
        End For2
    End For1
    Reduce h
End While
```

This is just the basic algorithm. There exists an implementation for constrained optimization: Implicit Filtering for constrained optimization (IFFCO). More details about that can be found in [7], [13]. There the authors also describe another feature of Implicit Filtering. It can use quasi-Newton updates to find a descent direction. This accelerates the convergence in the terminal phase of the iteration. We do not go into these details, but they document the successful application of this algorithm for noisy functions.

5.2 Classical results

The theoretical convergence results that have been achieved so far for the basic version are presented next. In Section 5.3 we compare them with the results we can produce by approaching this topic from another point of view.

Before we state the central theorem, we have to introduce some notational conventions. If we have a vector $x \in \mathbb{R}^n$ and a positive scalar $h \in \mathbb{R}^+$, then we define

$$S(x,h) := S(x, x+he_1, x+he_2, \ldots, x+he_n) \tag{5.2}$$

to be the simplex generated by $\{x, x+he_1, \ldots, x+he_n\}$, see Section 4.2. We also recall the supremum norm for functions $\varphi : \mathbb{R}^n \to \mathbb{R}$ on a set $S \subset \mathbb{R}^n$

$$\|\varphi\|_S = \sup_{x \in S} |\varphi(x)| \ .$$

Now we are ready to take a look at the classical convergence result for Implicit Filtering. It is classical in so far as it requires the existence of the gradient of the smooth function f_s. In the following theorem we assume the central simplex gradient to be the used approximation within the algorithm like Kelley does in [19].

There also exists a version using the forward simplex gradient, which employs the same ideas and techniques for the proof. A corresponding theorem can be found in [8]. However, in this case we cannot use the additional stencil failure test. On the other hand the differentiability assumption can be lowered. Only the Lipschitz continuity of ∇f_s is needed. But we reduce this restriction in Section 5.3 anyway to local Lipschitz continuity of f_s.

Theorem 5.1.
Let f_c satisfy (5.1), be bounded below and $\nabla^2 f_s$ be Lipschitz continuous on $\mathbb{R}^n$. In addition let $\{h_k\}_{k=1}^{\infty}$ be a sequence of scales such that $h_k \downarrow 0$, $\{x_k\}_{k=1}^{\infty}$ be the sequence produced by Implicit Filtering using a central difference simplex gradient approximation on $S_k = S(x_k, h_k)$. Let $R_k = R(S_k)$ denote the reflected simplex and $U_k := S_k \cup R_k$. If we assume that there are at most finitely many line search failures and

$$\lim_{k\to\infty}\left(h_k^2 + \frac{\|\varphi\|_{U_k}}{h_k}\right) = 0 \tag{5.3}$$

holds, then any limit point of $\{x_k\}_{k=1}^{\infty}$ is a critical point of f_s, i.e.

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

Proof:
Since the previously described algorithm involves numerous termination criteria, we have to take care of several cases. We show that in any case the gradient converges to zero.

i) infinitely often stencil failure

There exists a subsequence $\{x_{k_j}\}_{j=1}^{\infty}$ such that for every element of this subsequence stencil failure occurs. In Example 4.4 we showed for $j \in \mathbb{N}$ that $\sigma_+(S_{k_j}) = h_{k_j}$ and $\kappa\big(V(S_{k_j})\big) = 1$. We want to use Theorem 4.13 and mention that the prerequisite (4.9) is satisfied for all $j \in \mathbb{N}$ with $\mu = \frac{1}{2}$. Thus we find

$$\forall\, j \in \mathbb{N}: \quad \|\nabla f_s(x_{k_j})\| \;<\; 2\,K\left(h_{k_j} + \frac{\|\varphi\|_{U_{k_j}}}{h_{k_j}}\right).$$

We note that (5.3) and $h_k \downarrow 0$ imply $\left(h_{k_j} + \frac{\|\varphi\|_{U_{k_j}}}{h_{k_j}}\right) \to 0$ to conclude

$$\|\nabla f_s(x_{k_j})\| \to 0\,.$$

ii) infinitely often $\|\mathbf{D_h f(x)}\| \le \tau \mathbf{h}$

Since in the theorem we specified the used finite-difference approximation to be the central simplex gradient, we know that $D_h f(x) = D^c\big(f_c, S(x,h)\big)$. The case we are now investigating guarantees the existence of a subsequence $\{x_{k_j}\}_{j=1}^{\infty}$ such that for every j

$$\|D^c(f_c, S_{k_j})\| \le \tau h_{k_j}$$

holds. By the assumption that $h_k \downarrow 0$, this implies

$$\|D^c(f_c, S_{k_j})\| \to 0 \ .$$

In *i)* we already observed that $\sigma_+(S_{k_j}) = h_{k_j}$ and $\kappa\big(V(S_{k_j})\big) = 1$. This enables us together with (5.3) and Theorem 4.11 to conclude

$$\|\nabla f_s(x_{k_j})\| \to 0 \ .$$

For the iterates that are not covered by the above two cases, there is only one possibility left.

iii) Always but finitely often the line search succeeds

A successful line search is the only case in which there is an updated iterate $x_{k+1} = x_k - \lambda\,\beta^{l_k} D^c(f_c, S_k)$ that differs from the old iterate. Thus we may assume without loss of generality that there are no other iterates between two successful line searches. Then there exists an $N \in \mathbb{N}$ such that $\forall\, k \geq N$ the line search succeeds. We denote the accompanying stepsize by $\lambda_k := \lambda_{l_k} = \lambda\,\beta^{l_k}$ to find

$$f_c(x_{k+1}) - f_c(x_k) < -\alpha\,\lambda_k \|D^c(f_c, S_k)\|^2 \ .$$

By the algorithm λ_k is bounded below by $\lambda\,\beta^{MAXCUT}$. This implies

$$-\frac{f_c(x_{k+1}) - f_c(x_k)}{\alpha\,\lambda\,\beta^{MAXCUT}} > \|D^c(f_c, S_k)\|^2 > 0 \ .$$

In analogy to the prove of Theorem 4.16, we now use the fact that $\{f_c(x_k)\}_{k \geq N}$ is monotonically decreasing and by assumption bounded below. This guarantees that the left hand side of the above equation is converging to zero and thus

$$\|D^c(f_c, S_k)\| \to 0 \ .$$

Like in *ii)* Theorem 4.11 and (5.3) finally yield

$$\|\nabla f_s(x_k)\| \to 0 \ .$$

In any of the above cases we deduced that the limit of the exact gradients is zero. Thus the sequence $\{x_k\}_{k=1}^{\infty}$ might be created by any collection of these cases. The limit of the corresponding gradients is always zero and our proof is finished. ■

Some comments about this theorem and its assumptions are necessary. First note that if we assume $h_k \downarrow 0$, then the conditions

$$\lim_{k\to\infty} \frac{\|\varphi\|_{U_k}}{h_k} = 0 \quad , \tag{5.4}$$

$$\lim_{k\to\infty} (h_k + \frac{\|\varphi\|_{U_k}}{h_k}) = 0 \quad \text{and} \tag{5.5}$$

$$\lim_{k\to\infty} (h_k^2 + \frac{\|\varphi\|_{U_k}}{h_k}) = 0 \tag{5.6}$$

are all equivalent. They describe a restriction on the possible noise. If the error introduced by φ is above the upper bound given by these formulas, then its influence is too big. A complete analysis of the behavior of the smooth part f_s is no longer possible. In the next section we encounter a modified version of (5.4), which is necessary for the analysis.

An obviously very restrictive requirement is the Lipschitz continuity of $\nabla^2 f_s$. So actually we want f_s to be more than just in $C^1(\mathbb{R}^n)$. However, there are applications that do not have objective functions with a Lipschitz continuous Hessian. Even more there exist cases where f_s is not differentiable at all. For those cases we would like to be able to state convergence theorems as well. Section 5.3 is dedicated to this goal and the results contained therein open the door for a new kind of numerical analysis for noisy and non-smooth functions.

The last remaining assumption of at most finitely many line search failures sounds reasonable. Otherwise the finite differences are no accurate approximations. Consequently the whole algorithm would have to fail. But who guarantees the sufficient accuracy of our estimates?

An attempt could be to put more restrictions like (5.4) – (5.6) on the error introduced by the noise φ. We then might be able to transfer the implemented version of the line search method back to the smooth case and use the results from Section 4.3. But we now show that this attempt has to fail. Thus, the assumption of only finitely many line search failures has to remain an assumption.

We would like to guarantee for a fixed iterate $x_k \in \mathbb{R}^n$ and all following iterates that no line search failure occurs. In the following argumentation we use the forward simplex gradient. But a similar reasoning holds for the central and backward simplex gradient too. At first we have to show for x_k and some $l_k \in \{1, \dots, \texttt{MAXCUT}\}$ that

$$f_c\big(x_k + \lambda\,\beta^{l_k} D^+(f_c, S_k)\big) - f_c(x_k) < -\alpha\,\lambda\,\beta^{l_k} \|D^+(f_c, S_k)\|^2 \tag{5.7}$$

is definitely true. To shorten the notation we define $\lambda_k = \lambda\,\beta^{l_k}$. By the definition of f_c , (5.7) is equivalent to

$$\begin{aligned} f_s(x_k + \lambda_k D^+(f_c, S_k)) - f_s(x_k) \quad < \quad & -\alpha\,\lambda_k \|D^+(f_c, S_k)\|^2 \\ & -\varphi(x_k + \lambda_k D^+(f_c, S_k)) + \varphi(x_k) \;. \end{aligned}$$

Since we want to get a sufficient condition involving f_s for this inequality, we use (4.5) twice to find that

$$\begin{aligned} f_s(x_k+\lambda_k\nabla f_s(x_k)) - f_s(x_k) \quad < \quad & -\alpha\,\lambda_k\|\nabla f_s(x_k)\|^2 \\ & -2\,\alpha\,\lambda_k\|\nabla f_s(x_k)\|K\Big(h_k+\frac{\|\varphi\|_{U_k}}{h_k}\Big) \\ & -\alpha\,\lambda_k K^2\Big(h_k+\frac{\|\varphi\|_{U_k}}{h_k}\Big)^2 \\ & -L\,\lambda_k K\Big(h_k+\frac{\|\varphi\|_{S_k}}{h_k}\Big) \\ & -\varphi\big(x_k+\lambda_k D^+(f_c,S_k)\big)+\varphi(x_k) \end{aligned}$$

has to be satisfied. Here L denotes the Lipschitz constant of f_s . Note that we again used the facts $\sigma_+(S_k)=h_k$ and $\kappa\big(V(S_k)\big)=1$. We want to make use of Theorem 4.15 and thus have to require for $\tilde{\alpha}:=1-\alpha$ that

$$\begin{aligned} -\tilde{\alpha}\,\lambda_k\|\nabla f_s(x_k)\|^2 \quad < \quad & -2\,\alpha\,\lambda_k\|\nabla f_s(x_k)\|K\Big(h_k+\frac{\|\varphi\|_{S_k}}{h_k}\Big) \\ & -\alpha\,\lambda_k K^2\Big(h_k+\frac{\|\varphi\|_{U_k}}{h_k}\Big)^2 \\ & -L\,\lambda_k K\Big(h_k+\frac{\|\varphi\|_{S_k}}{h_k}\Big) \\ & -\varphi\big(x_k+\lambda_k D^+(f_c,S_k)\big)+\varphi(x_k)\ . \end{aligned}$$

We now define $\delta_k=\lambda_k\|D^+(f_c,S_k)\|$ and denote the ball around x_k with radius δ_k by $B_{\delta_k}(x_k)$. Using for the supremum norm of φ to get an estimate for the last line, we find (multiplying the above with -1) our final inequality

$$\begin{aligned} \tilde{\alpha}\,\lambda_k\|\nabla f_s(x_k)\|^2 \quad > \quad & 2\,\alpha\,\lambda_k\|\nabla f_s(x_k)\|K\Big(h_k+\frac{\|\varphi\|_{U_k}}{h_k}\Big) \\ & +\alpha\,\lambda_k K^2\Big(h_k+\frac{\|\varphi\|_{U_k}}{h_k}\Big)^2 \\ & +L\,\lambda_k K\Big(h_k+\frac{\|\varphi\|_{U_k}}{h_k}\Big) \\ & +2\|\varphi\|_{B_{\delta_k}(x_k)}\ . \end{aligned}$$

Assume we could require h_k, $\|\varphi\|_{U_k}$ and $\|\varphi\|_{B_{\delta_k}(x_k)}$ to decrease fast enough in relation to $\|\nabla f_s(x_k)\|$. Then the above inequality would be true for x_k and all the following iterates. That this problem can be solved in a sensible way is questionable. Even if it were solved, then we would be in the case of Theorem 4.15. It only guarantees that for some $\lambda>0$ Equation (5.7) holds. But we cannot be sure that one of our finitely many $\lambda_k=\lambda\,\beta^{l_k}$ ($l_k\in\{1,\dots,\texttt{MAXCUT}\}$) is small enough to be one of them. And in order to keep the Implicit Filtering algorithm well defined, we have to insist on testing only finitely many l_k.

However, there are theorems that provide intervals for every iteration, from which the stepsize can be chosen to find sufficient decrease. But upper and lower bounds of those intervals depend on $\|\nabla f_s(x_k)\|$ and the Lipschitz constant of ∇f_s. Since none of them is known to us, the assumption: "*There are at most finitely many line search failures*", has to remain an assumption. We now turn our attention to another requirement that can be relaxed. To get rid of the necessity of smoothness for f_s, we use the methods and techniques introduced in the Chapters 2 and 3.

5.3 Nonsmooth convergence analysis

In this section we relax the requirement of Fréchet differentiability for the convergence analysis. To do so we use set-valued mappings and the necessary condition for the Clarke derivative of Theorem 3.10. Since we only have to assume local Lipschitz continuity for the function we denoted so far by f_s, we change the notation to f_L.

First, we have to justify the set-valued approach. For this purpose remember that the function f_c that we can compute is of the form

$$\forall\, x \in \mathbb{R}^n : \quad f_c(x) = f_L(x) + \varphi(x)\ . \tag{5.8}$$

Also the bounded function φ is allowed to produce different outcomes when evaluated twice with the same argument. Consequently the value $f_c(x)$ is somehow randomly distributed. We only know that it lies somewhere within an interval around $f_L(x)$. The size of this interval is determined by the bounds of φ. If we now think of this interval to be a part of the image set of an envelope mapping around f_L, then $f_c(x)$ is an arbitrary element of this image set. To be more precise let F_{f_L} be an envelope mapping around f_L given by (3.1), then we require

$$\forall\, x \in \mathbb{R}^n : \quad f_c(x) \in F_{f_L}(x) = \{y \in \mathbb{R} \,|\, f_-(x) \le y \le f_+(x)\}\ .$$

The following example illustrates this thought.

Example 5.2.
Figure 5.1 on the next page displays several perturbations of the function $f_L : \mathbb{R} \to \mathbb{R}$ defined by $f_L(x) := x^2$. They are all of the kind described in [13]. We plotted their graphs on top of an envelope mapping (grey area). Note that its image set contains the entire graphs of all of the noisy functions. If we only know about a computed function value $f_c(x)$ that it is a value of one of the perturbations, then we know it is an element of the image set of the envelope mapping and vice versa. $\square$

The next lemma prepares us for the proof of the main convergence theorem. It tells us how to draw conclusions about the existence of critical points of f_L , if we only have information about its envelope mapping. Especially, we need information about the lower bound f_- . As in the previous section, we recognize that the magnitude of the possible noise is crucial for this step. It must not be too large.

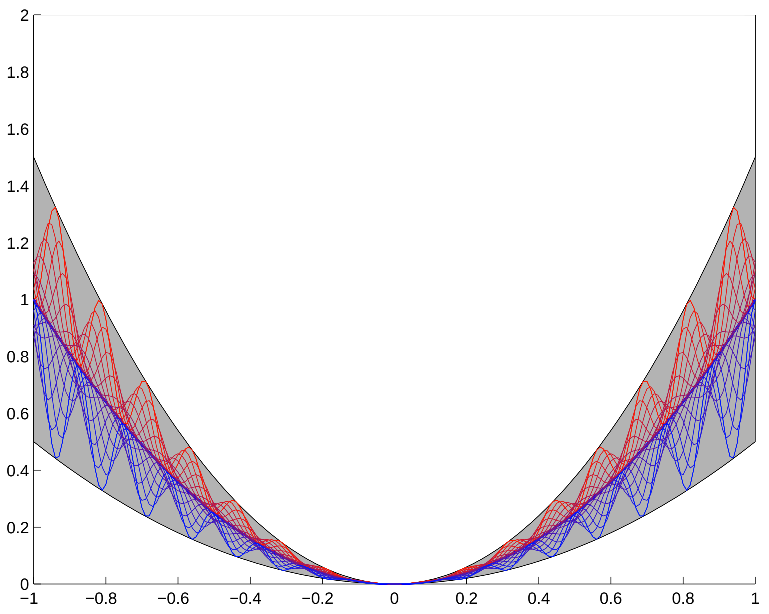

Figure 5.1: Perturbations and envelope mapping of a quadratic function

Lemma 5.3.
Let $x \in \mathbb{R}^n$ and $f_-, f_L, f_+ : \mathbb{R}^n \to \mathbb{R}$ such that f_L is Lipschitz continuous around x. Let F_{f_L} be the envelope mapping around f_L given by (3.1) and $\{p_i\}_{i=1}^m$ be a positive spanning set of $\mathbb{R}^n$. In addition let $\{x_k\}_{k=1}^\infty \subset \mathbb{R}^n$ with $x_k \to x$ and $\{h_k\}_{k=1}^\infty \subset \mathbb{R}^+$ with $h_k \downarrow 0$ such that

i) $\forall\, i \in \{1, \dots, m\} : \limsup\limits_{k\to\infty} \frac{f_-(x_k+h_k p_i)-f_-(x_k)}{h_k} \geq 0$ *and*

ii) $\lim\limits_{k\to\infty} \frac{f_+(x_k)-f_-(x_k)}{h_k} = 0$

hold, then

$$\forall\, d \in \mathbb{R}^n : \quad Df_L(x,d) \geq 0\,.$$

Proof:
Since f_L is Lipschitz continuous around x, we know by Equation (2.9) that

$$\forall\, d \in \mathbb{R}^n : \quad Df_L(x,d) = \limsup_{\substack{x_k\to x\\ h_k\downarrow 0}} \frac{f_L(x_k+h_k d) - f_L(x_k)}{h_k}\,.$$

We use this equation to observe for any p_i ($i \in \{1, \dots, m\}$) that

$$
\begin{aligned}
Df_L(x,p_i) &= \limsup_{\substack{x_k\to x\\ h_k\downarrow 0}} \frac{f_L(x_k+h_kp_i)-f_L(x_k)}{h_k}\\
&\geq \limsup_{k\to\infty} \frac{f_L(x_k+h_kp_i)-f_L(x_k)}{h_k}\\
&\overset{f_+\geq f_L}{\geq} \limsup_{k\to\infty} \frac{f_-(x_k+h_kp_i)-f_-(x_k)+f_-(x_k)-f_+(x_k)}{h_k}\\
&= \limsup_{k\to\infty} \Big[\frac{f_-(x_k+h_kp_i)-f_-(x_k)}{h_k} - \frac{f_+(x_k)-f_-(x_k)}{h_k}\Big]\\
&\overset{i),ii)}{\geq} 0\ .
\end{aligned}
$$

But $\{p_i\}_{i=1}^m$ is a positive spanning set of $\mathbb{R}^n$, i.e.

$$
\forall\, d\in\mathbb{R}^n\ \ \exists\,\{\lambda_i\}_{i=1}^m\subset\mathbb{R}_0^+ : \quad d=\sum_{i=1}^m \lambda_i p_i\ .
$$

The sublinearity of $Df_L(x,\cdot)$, compare (2.1) and Theorem 2.22, now yields

$$
\begin{aligned}
\forall\, d\in\mathbb{R}^n : \quad Df_L(x,d) &= Df_L\Big(x,\sum_{i=1}^m\lambda_ip_i\Big)\\
&\geq \sum_{i=1}^m \lambda_i Df_L(x,p_i)\geq 0\ .
\end{aligned}
$$

■

The last lemma showed us that f_- is not only closely related to F_{f_L}. If the allowed tolerance for the values of F_{f_L} around f_L is small enough, then we can also draw conclusions about f_L itself. Condition *i)* of the lemma implies on the one hand for Lipschitz continuous f_- that $Df_-(x,p_i)\geq 0$ and thus $DF_{f_L}(x,p_i)\geq 0$ for all $i\in\{1,\dots,m\}$. On the other hand we have shown that it implies in conjunction with *ii)* also $Df_L(x,p_i)\geq 0$. Using the sublinearity of the Clarke derivative we finally generalized the last inequality for all vectors $d\in\mathbb{R}^n$. Thus, the concepts of an envelope mapping around a function and the Clarke derivative for envelope mappings work together hand in hand for this proof.

In order to use Lemma 5.3 in the prove of our convergence result, we need a positive spanning set of $\mathbb{R}^n$. However, for this purpose we have to modify the basic algorithm we stated in Section 5.1 slightly. We cannot simply use one of the simplex gradient approximations as we did in the previous section. We have to use a combined version of the forward and backward simplex approximation. The basic idea is to average them. This leads within the algorithm to the use of

$$\frac{1}{2}(\|D^{+}(f,S)\| + \|D^{-}(f,S)\|)$$

instead of just one of the norms. We also use the average of both approximations as anticipated descent direction. This is by Definition 4.5 the central simplex gradient, which appears in the sufficient decrease condition of the algorithm below.

Algorithm: Modified Implicit Filtering

```
Set MINH, MAXH as limits for the scales H
Set MAXIT for the admissible number of line search iterations
Set MAXCUT for the admissible number of line search backtracking steps
Set $\alpha$, $\beta$, $\lambda$ as additional line search parameters
Set $\tau$ as termination tolerance for $\frac{1}{2}(\|D^{-}f(x,S)\| + \|D^{+}f(x,S)\|) \leq \tau \cdot h$

Initialize the starting point x
Initialize the first scale h=MAXH

While h $\geq$ MINH do
    For1 m = 1,...,MAXIT
        Compute the simplex gradients $D^{-}f(x,S), D^{+}f(x,S)$
        If stencil failure occurs:  Exit For1
        If $\frac{1}{2}(\|D^{-}f(x,S)\| + \|D^{+}f(x,S)\|) \leq \tau \cdot h$ :  Exit For1
        For2 l = 0,...,MAXCUT
            Set $\lambda_l = \lambda\beta^l$
            If $f(x - \lambda_l D^c f(x,S)) - f(x) < -\frac{\alpha}{2}\lambda_l (\|D^{-}f(x,S)\|^2 + \|D^{+}f(x,S)\|^2)$ :
                Update $x = x - \lambda_l D^c f(x,S)$
                Exit For2
            Else:  If l = MAXCUT: Exit For1
        End For2
    End For1
    Reduce h
End While
```

In the above algorithm we used the symbol S for the simplex defined in Equation (5.2). It is also specified more precisely in the following convergence theorem.

Theorem 5.4.
Let $f_-, f_L, f_+ : \mathbb{R}^n \to \mathbb{R}$ such that f_- is bounded below on $\mathbb{R}^n$ and let F_{f_L} be the envelope mapping around f_L given by (3.1). In addition let $\{h_k\}_{k=1}^{\infty}$ be a sequence of scales such that $h_k \downarrow 0$ and $\{x_k\}_{k=1}^{\infty}$ be the sequence produced by the modified Implicit Filtering algorithm using simplex gradient approximations on $S_k = S(x_k, h_k)$ and computed function values $f_c(\cdot) \in F_{f_L}(\cdot)$. If we assume that there are at most finitely many line search failures and

$$\lim_{k\to\infty} \frac{\|f_+ - f_-\|_{S_k \cup R(S_k)}}{h_k} = 0 \tag{5.9}$$

holds, then any limit point x_* of $\{x_k\}_{k=1}^{\infty}$, around which f_L is Lipschitz continuous, is a critical point of f_L, i.e.

$$\forall\, d \in \mathbb{R}^n : \quad Df_L(x_*, d) \geq 0 \,.$$

Proof:
Let x_* be a limit point of $\{x_k\}_{k=1}^{\infty}$ around which f_L is Lipschitz continuous. Then there exists a subsequence $\{x_{k_j}\}_{j=1}^{\infty} =: \{x'_j\}_{j=1}^{\infty}$ converging to x_0. We denote the corresponding subsequence of $\{h_k\}_{k=1}^{\infty}$ by $\{h'_j\}_{j=1}^{\infty}$. We want to use Lemma 5.3, so we check its prerequisites. The functions f_-, f_L, f_+ are already defined the way we need them. The positive spanning set is given by the simplex directions of S_k and $R(S_k)$. It is the set of unit vectors $\{e_1, -e_1, \ldots, e_n, -e_n\} =: P$. We also know that $x'_j \to x_*$ and $h'_j \downarrow 0$. Also condition *ii)* of the lemma is satisfied, because of (5.9). The only thing we are left with to show is

$$\forall\, p \in P : \quad \limsup_{j\to\infty} \frac{f_-(x'_j + h'_j p) - f_-(x'_j)}{h'_j} \geq 0 \,.$$

This is equivalent to

$$\forall\, p \in P \;\; \forall\, \varepsilon > 0 \;\; \forall\, N \in \mathbb{N} \;\; \exists\, j \geq N : \quad \frac{f_-(x'_j + h'_j p) - f_-(x'_j)}{h'_j} > -\varepsilon \,.$$

Thus let $p \in P$, $\varepsilon > 0$, $N \in \mathbb{N}$ be arbitrary. Since there are only finitely many line search failures, we can find an $M \in \mathbb{N}$ such that for all $j \geq M$ one of the following cases occurs:

i) $f_c(x'_j + h'_j p) \geq f_c(x'_j)$,

ii) $\frac{1}{2}\left(\|D^- f_c(x'_j, S'_j)\| + \|D^+ f_c(x'_j, S'_j)\|\right) \leq \tau\, h'_j$ or

iii) $f_c(x'_{j+1}) - f_c(x'_j) < -\frac{\alpha}{2}\, \lambda_{l'_j} \left(\|D^- f_c(x'_j, S'_j)\|^2 + \|D^+ f_c(x'_j, S'_j)\|^2\right)$.

We examine these cases separately.

i) stencil failure

Since $f_+ \geq f_c \geq f_-$ holds, we find

$$\begin{aligned} \frac{f_-(x'_j + h'_j p) - f_-(x'_j)}{h'_j} &\geq \frac{f_-(x'_j + h'_j p) - f_c(x'_j)}{h'_j} \\ &\geq \frac{f_-(x'_j + h'_j p) - f_+(x'_j + h'_j p)}{h'_j} \\ &\quad + \frac{f_c(x'_j + h'_j p) - f_c(x'_j)}{h'_j} \\ &\geq \frac{f_-(x'_j + h'_j p) - f_+(x'_j + h'_j p)}{h'_j} \\ &\geq -\frac{\varepsilon}{2} > -\varepsilon \, . \end{aligned}$$

Because of (5.9), there exist a constant $M_1 \in \mathbb{N}$ such that the last line of the above inequalities is true for all $j \geq M_1$.

ii) termination criterion

By the definition of the Euclidean norm we can find constants $K, M_2 \in \mathbb{N}$ such that

$$\frac{f_c(x'_j + h'_j p) - f_c(x'_j)}{h'_j} \geq -K h'_j > -\frac{\varepsilon}{2}$$

holds for all $j \geq M_2$. Like in *i)* we can use Equation (5.9), to conclude for all $j \geq \max\{M_1, M_2\}$

$$\frac{f_-(x'_j + h'_j p) - f_-(x'_j)}{h'_j} > -\varepsilon \, .$$

iii) successfull line search

We observe by looking at the algorithm that the sequence $\{f_c(x'_j)\}_{j=1}^{\infty}$ is monotonically decreasing. It is bounded below because of the boundedness of f_- . Thus the sequence converges and we find $|f_c(x'_{j+1}) - f_c(x'_j)| \to 0$. The successful line search implies

$$(\|D^- f_c(x'_j, S'_j)\|^2 + \|D^+ f_c(x'_j, S'_j)\|^2) < -\frac{2(f_c(x'_{j+1}) - f_c(x'_j))}{\alpha \, \lambda_{l'_j}} \, .$$

The algorithm also guarantees the boundedness of $\lambda_{l'_j}$. Like in *ii)* we can find constants $\tilde{K}, M_3 \in \mathbb{N}$ such that

$$\frac{f_c(x'_j + h'_j p) - f_c(x'_j)}{h'_j} \geq \tilde{K}(f_c(x'_{j+1}) - f_c(x'_j)) > -\frac{\varepsilon}{2}$$

holds for all $j \geq M_3$. Similarly to *ii*) we find for all $j \geq \max\{M_1, M_3\}$

$$\frac{f_-(x'_j + h'_j p) - f_-(x'_j)}{h'_j} \quad > \quad -\varepsilon \ .$$

Now we pick any $j \geq \max\{N, M, M_1, M_2, M_3\}$ and conclude that

$$\frac{f_-(x'_j + h'_j p) - f_-(x'_j)}{h'_j} > -\varepsilon \ .$$

Consequently

$$\limsup_{j\to\infty} \frac{f_-(x'_j + h'_j p) - f_-(x'_j)}{h'_j} \geq 0$$

and we can apply Lemma 5.3. This yields the desired result. ■

To see the power of this theorem and to compare it with Theorem 5.1, we make some concluding remarks.

The applications of the algorithm are often very complex. Because of this it can be assumed that the main computational expense is caused by the evaluations of the objective function. The modified Implicit Filtering algorithm and the basic version using the central simplex gradient require the same number of function evaluations. Thus both would be approximately of the same expense.

The advantage of the modified algorithm lies in the fact that Theorem 5.4 now guarantees for any locally Lipschitz continuous function to produce a critical point in the case of convergence. The basic version with the central simplex gradient only provides theoretical results for functions in $C^2(\mathbb{R}^n)$, which is a much smaller class of functions.

But there is a weakness of the critical point produced by the modified algorithm. We showed in Section 3.4 that the necessary condition involving the Clarke derivative is not as strong as the gradient based version. However, we also saw that it is not easy to find examples, for which this failure occurs.

Chapter 6

Pattern Search Methods

The next class of algorithms we examine are Pattern Search algorithms. Torczon [26] introduced the term General Pattern Search (GPS) for a class of algorithms that contains for example the *Pattern Search* algorithm of Hooke and Jeeves [17] or the *Multidirectional Search* algorithm of Dennis and Torczon [12]. In contrast to Implicit Filtering and other gradient based methods, they do not compute or explicitly approximate any derivatives. So actually these methods are a special kind of *direct search* algorithms. Their common basis is the way they perform their search for a better iterate. It is carried out on a pattern of points that is independent of the objective function f.

Like in the previous chapter we first give a brief description of the basic algorithm. Doing so, we mostly follow [26] in Section 6.1. Then we present some of the known convergence theory in Section 6.2, starting with the purely gradient based results of Torczon. In this section we also have to talk about recent results by Audet and Dennis [2], [3], who have already introduced the Clarke derivative in the context of GPS algorithms (see Subsection 6.2.2). Although originally this generalized derivative concept and Pattern Search methods were developed independent of each other, the two concepts seem to harmonize and work together very well. We see this especially in Subsection 6.2.3. There we go one step further than Audet and Dennis and use our necessary condition from Theorem 3.10. This is the first new result of this chapter and in contrast to [2], totally free of any classical differentiation assumption. Crucial for our innovative results is – like in Chapter 5 – the sublinearity of the Clarke derivative. To our knowledge, this feature has not been recognized and exploited in this context so far. In addition we extend the convergence theory to the case of noisy functions. Like in Chapter 5 we use the Clarke derivative of an envelope mapping that surrounds the inexact function values.

6.1 The Basic Algorithm

Since these methods do not depend on any derivative information, they are predestined for minimization problems

$$\min_{x \in \mathbb{R}^n} f(x) \tag{6.1}$$

where this information is hard to obtain or not to obtain at all. A practical example for such a situation is the optimal design of a thermal insulation system as it is described in [1]. To explain how GPS works, we have to answer two main questions:

1. What is the so-called pattern and how is it generated?
2. How do we have to proceed on this pattern to be successful in finding a critical point of (6.1)?

Both tasks are described in detail in [26]. From there we outline the important facts. To answer the first question, we have to define several matrices.

Definition 6.1.
Let $\mathcal{M} \subset \mathbb{Z}^{n,n}$ be a finite set of nonsingular matrices, $M_k \in \mathcal{M}$ and $L_k \in \mathbb{Z}^{n,m}$. Then for $p := 2n + m$ the matrix

$$G_k := \begin{bmatrix} M_k & -M_k & L_k \end{bmatrix} = \begin{bmatrix} \Gamma_k & L_k \end{bmatrix} \in \mathbb{Z}^{n,p} \tag{6.2}$$

is called a *generating matrix.*

Any nonsingular matrix $B \in \mathbb{R}^{n,n}$ is called a *basis matrix.*

The set of columns of the matrix

$$P_k := BG_k = \begin{bmatrix} B\Gamma_k & BL_k \end{bmatrix} \in \mathbb{R}^{n,p} \tag{6.3}$$

is called a *pattern.* $\square$

Note that the column vectors of $B\Gamma_k$, and thus the entire pattern, form a positive spanning set of $\mathbb{R}^n$, because B and M_k are both nonsingular. This observation is important for the convergence analysis. The matrix L_k is not important for the theory. In the application it reflects a heuristic search method based on some experience with the practical problem. It is designed to take advantage of specific knowledge about the particular situation and to accelerate the computational progress.

The pattern determines the directions in which one is allowed to search for new iterates. How this has to be done is the next question. In order to be able to answer it, we have to make some more definitions.

Definition 6.2.
Let $B \in \mathbb{R}^{n,n}$ be a basis matrix, $G_k \in \mathbb{Z}^{n,p}$ be a generating matrix and for $i = 1, \ldots, p$ let g_k^i denote the column vectors of G_k. Then for a positive $h_k \in \mathbb{R}$ the vectors s_k^i defined by

$$\forall\, i \in \{1, \ldots, p\} : \quad s_k^i := h_k B g_k^i \tag{6.4}$$

are called *trial steps* with *step size parameter* h_k and *direction* Bg_k^i.

For an iterate $x_k \in \mathbb{R}^n$ any point of the form

$$x_k^i := x_k + s_k^i \tag{6.5}$$

is called a *trial point*. □

The search on the pattern is performed by carrying out so-called *exploratory moves* about the current iterate x_k. These moves test new iterates of the form $x_{k+1} = x_k + s_k$ to find a lower function value. How this procedure is done exactly depends on the respective special algorithm. But there are some basic rules that must be obeyed. They are

1. s_k has to be of the form (6.4) and
2. If $\min\limits_{i \in \{1,\ldots,2n\}} f(x_k + h_k B g_k^i) < f(x_k)$, then $f(x_{k+1}) = f(x_k + s_k) < f(x_k)$.

The first rule is the more precise form of saying that the search is only performed on a special pattern of points. To understand the second rule one has to recognize that the minimum is only taken among the columns of $h_k B \Gamma_k$. It means if we can find a simple decrease in one of the directions of our positive spanning set, then the new iterate must lead to an improvement in the function value. Or formulated indirectly: If the new iterate brings no improvement, then

$$\forall\, i \in \{1, \ldots, 2n\} : \quad f(x_k) \le f(x_k + h_k B g_k^i) \ .$$

In Chapter 4 we called such a situation a stencil failure, compare Definition 4.12. In order to be able to state a GPS algorithm we need to know one more detail. How are the values of x_k, h_k and G_k updated from one iteration to the next iteration? We have to distinguish two cases depending on the success of the pattern search.

If we find a better iterate $x_k + s_k$ on the pattern, then we update $x_{k+1} = x_k + s_k$. Otherwise the current iterate remains unchanged. The update of G_k depends on the pattern search method. It only has to follow Definition 6.1, i.e. $G_k \in \mathbb{Z}^{n,p}$ and the submatrices M_k must be chosen from a finite set of nonsingular matrices. The change of the step length parameter is a little more complicated. In general we can say that h_k has to be reduced, if the search for a better iterate fails. Otherwise it stays unchanged or is increased. The exact procedure is as follows. Given $\tau \in \mathbb{Q}$ with $\tau > 1$, $L \in \mathbb{N}$ and $\{\omega_i\}_{i=0}^{L} \subset \mathbb{Z}$ such that $\omega_0 < 0 \leq \omega_1 \leq \omega_2 \leq \cdots \leq \omega_L$, we update for some $i \in \{1, \ldots, L\}$

$$h_{k+1} = \begin{cases} \tau^{\omega_i} h_k \ , & \text{if } \ f(x_k + s_k) < f(x_k) \ \text{ for one } \ s_k \\ \tau^{\omega_0} h_k \ , & \text{otherwise} \end{cases} . \tag{6.6}$$

Collecting all the presented details together, we are able to state a GPS algorithm.

Algorithm: General Pattern Search

```
Set $\mathcal{M}$ the finite set of nonsingular integer matrices
Set B the basis matrix
Set $\tau$, $\omega_0, \ldots, \omega_L$ the parameters for the step length update

Choose $M_0 \in \mathcal{M}$ and an $L_0$
Initialize the pattern matrix $P_0$
Initialize the starting point $x_0$
Initialize the first step length $h_0$

For $k = 0, 1, \ldots$
    Compute the function value $f(x_k)$
    Determine a step $s_k$ from the set of trial steps
    If $f(x_k + s_k) < f(x_k)$ :
        Update $x_{k+1} = x_k + s_k$
        Update $h_{k+1} = \tau^{\omega_i} h_k$ for some $i \in \{1, \ldots, L\}$
        Update $G_k$
    Else
        Update $x_{k+1} = x_k$
        Update $h_{k+1} = \tau^{\omega_0} h_k$
        Update $G_k$
End For
```

Based on this framework we can now start to examine the various possible convergence theorems.

6.2 Convergence Analysis

Our convergence analysis starts with some important lemmas. Their proofs can be found in [26]. From there we also cite the main convergence theorem of Torczon. It assumes smoothness for the function f in Equation (6.1) on a neighborhood of the entire investigated domain. Next we turn our attention to the results of Audet and Dennis [2]. They were able to relax this assumption drastically to strict differentiability only at special limit points by incorporating the concept of the Clarke derivative. After that we present our own results. They do not depend on any differentiability assumption. We only need to require the Lipschitz continuity of the objective f.

6.2.1 Gradient based analysis

The first lemma we present is a justification for the name step length parameter for h_k.

Lemma 6.3.
Let $s_k^i \in \mathbb{R}^n \setminus \{O\}$ be a trial step produced by a GPS method, then there exists a constant $c > 0$ such that

$$\|s_k^i\| \geq c \cdot h_k \ .$$

Thus, h_k prevents the algorithm from making excessively short steps. Note that under additional assumptions we could formulate the inequality in the other direction as well. What we would need is an upper bound for $\|g_k^i\|$. Since the set of integer matrices $\mathcal{M}$ is finite, we only have to worry about the columns of L_k. They have to be bounded above. Torczon needs this other inequality to prove stronger convergence results, see [26, Thm 3.7].

The next lemma reveals an interesting fact about the structure of the iterates.

Lemma 6.4.
Let $x_0 \in \mathbb{R}^n$ be the starting point of a GPS method, h_0 the initial step length parameter and B the basis matrix. Let $\alpha, \beta \in \mathbb{N}$ such that $\tau = \frac{\alpha}{\beta}$ and x_k be the k^{th} iterate produced by GPS, then there exist $r_k, s_k \in \mathbb{Z}$ and $z_i \in \mathbb{Z}^n$ $(i = 0, \ldots, k-1)$ such that

$$x_k = x_0 + \frac{\alpha^{r_k}}{\beta^{s_k}} \, h_0 \, B \, \sum_{i=0}^{k-1} z_i \ . \tag{6.7}$$

This means that for a fixed k all iterates lie on an integer grid around x_0 generated by the columns of $\alpha^{r_k} \, \beta^{-s_k} \, h_0 \, B$. This structural result can be used to prove the following lemma.

Lemma 6.5.
Let $x_0 \in \mathbb{R}^n$ be the starting point for a GPS method and h_0 the initial step length parameter. Let $f : \mathbb{R}^n \to \mathbb{R}$ be such that $L(x_0) := \{x \in \mathbb{R}^n \,|\, f(x) \le f(x_0)\}$ is bounded and $\{h_k\}_{k=0}^{\infty}$ be the sequence of step length parameters produced by the GPS method for f, then

$$\liminf_{k\to\infty} h_k = 0\ . \tag{6.8}$$

This last result can be strengthened. If we assume $\{h_k\}_{k=0}^{\infty}$ to be nonincreasing, then it is an easy corollary to show that

$$\lim_{k\to\infty} h_k = 0\ . \tag{6.9}$$

We can achieve this by choosing $\omega_0 = 0$ for the update of h_k.

Note that so far the only assumption we had to make about the objective function f was the boundedness of $L(x_0) := \{x \in \mathbb{R}^n \,|\, f(x) \le f(x_0)\}$. However, in order to get the following convergence theorem, compare [26, Thm. 3.5], from Lemma 6.5 Torczon has to impose much stronger restrictions on the objective.

Theorem 6.6.
Let $x_0 \in \mathbb{R}^n$ be the starting point for a GPS method. Let $f : \mathbb{R}^n \to \mathbb{R}$ be such that $L(x_0) := \{x \in \mathbb{R}^n \,|\, f(x) \le f(x_0)\}$ is compact and f is continuously differentiable on a neighborhood of $L(x_0)$, then we find for the sequence of iterates $\{x_k\}_{k=0}^{\infty}$ produced by the GPS method with objective f that

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

We want to mention that by restricting the algorithm, one can obtain the stronger result

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

Among other things we would have to assure the already mentioned boundedness of the columns of the generating matrices G_k and (6.9). Looking back at the listed lemmas and theorems, it seems unsatisfactory to restrict f that much just to be able to do the last step of the convergence analysis. That is why we turn our attention now to the work of Audet and Dennis.

6.2.2 Mixed analysis

In [2] Audet and Dennis already employ the concept of the Clarke derivative. But they do not use it to the full extent. In the end they prove a necessary condition that is again based on the gradient of the objective function. In this Subsection we present their argumentation. At first they define a refining subsequence.

Definition 6.7.
Let $\{x_k\}_{k=0}^{\infty}$ be a sequence of iterates produced by a GPS method, $\{h_k\}_{k=0}^{\infty}$ the corresponding sequence of step length parameters and $J \subset \mathbb{N}$ be an infinite set of indices. Then a subsequence $\{x_j\}_{j\in J}$ with the properties

i) $\lim\limits_{\substack{j\to\infty \\ j\in J}} h_j = 0$ and

ii) $\exists\, x_* := \lim\limits_{\substack{j\to\infty \\ j\in J}} x_j$

is called a *refining subsequence.* □

In fact, if we assume the set

$$L(x_0) := \{x \in \mathbb{R}^n \,|\, f(x) \le f(x_0)\} \tag{6.10}$$

to be bounded, then it is easy to prove the existence of a refining subsequence.

Lemma 6.8.
Let $x_0 \in \mathbb{R}^n$ be the starting point for a GPS method. Let $f : \mathbb{R}^n \to \mathbb{R}$ be such that $L(x_0) := \{x \in \mathbb{R}^n \,|\, f(x) \le f(x_0)\}$ is bounded. If $\{x_k\}_{k=0}^{\infty}$ is the sequence of iterates produced by the GPS method with objective f, then there exists a refining subsequence of $\{x_k\}_{k=0}^{\infty}$.

Proof:
By Lemma 6.5 we know that there exists a subsequence of the step length parameters $\{h_k\}_{k=0}^{\infty}$ converging to zero. Since the corresponding iterates are in the bounded set $L(x_0)$, the theorem of Bolzano & Weierstraß guarantees the existence of a converging subsequence. This is obviously a refining subsequence. ■

Next we state and prove a theorem that is close to [2, Thm 3.5]. The introduced modifications allow us to make use of this result in Subsection 6.2.3.

Theorem 6.9.
Let $f : \mathbb{R}^n \to \mathbb{R}$ and x_* be the limit point of a refining subsequence of a GPS method with objective f. If f is Lipschitz continuous around x_*, then there exists a positive spanning set $\{p_i\}_{i=1}^{2n}$ of $\mathbb{R}^n$ such that

$$\forall\, i \in \{1, \dots, 2n\} : \quad Df(x_*, p_i) \ge 0\,. \tag{6.11}$$

Proof:
Let x_* be the limit point of a refining subsequence, i.e.

$$x_* = \lim_{\substack{j\to\infty \\ j\in J}} x_j \qquad \text{and} \qquad \lim_{\substack{j\to\infty \\ j\in J}} h_j = 0\,. \tag{6.12}$$

Since the step length parameter is reduced only if an iteration is unsuccessful, we may assume without loss of generality that $\{x_j\}_{j\in J}$ is a subsequence of unsuccessful iterations. But this means for the trial steps denoted by s_j^i

$$\forall\, j \in J\ ,\ i \in \{1,\dots,2n\}: \quad f(x_j) \le f(x_j + s_j^i)\ . \tag{6.13}$$

Remember that by Definition 6.2 the set $\{s_j^i\}_{i=1}^{2n}$ is a positive spanning set of $\mathbb{R}^n$ for all $j \in J$. Furthermore, since the set $\mathcal{M}$ of Definition 6.1 is finite, at least one of them occurs infinitely often for $j \in J$. We denote this positive spanning set by $\{p_i\}_{i=1}^{2n}$ and the corresponding subset of indices by $\tilde{J}$. Now the additionally assumed Lipschitz continuity of f around x_* yields

$$\begin{aligned}\forall\, i \in \{1,\dots,2n\}: \quad Df(x_*, p_i) &\overset{(2.9)}{=} \limsup_{\substack{x_k \to x_* \\ h_k \downarrow 0}} \frac{f(x_k + p_i) - f(x_k)}{h_k} \\ &\overset{(6.12)}{\ge} \limsup_{\substack{j \to \infty \\ j \in J}} \frac{f(x_j + p_i) - f(x_j)}{h_j} \\ &\overset{(6.13)}{\ge} 0\ .\end{aligned}$$

■

Based on this theorem Audet and Dennis are able to show that the limit point of a refining subsequence satisfies the following necessary condition.

Theorem 6.10.
Let $x_0 \in \mathbb{R}^n$ be the starting point for a GPS method and $f : \mathbb{R}^n \to \mathbb{R}$ be such that $L(x_0)$ is bounded. Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of iterates produced by the GPS method with objective f and let x_* be the limit of a refining subsequence. If f is strictly differentiable at x_*, then

$$\nabla f(x_*) = O\ .$$

Proof:
First of all we want to mention that the above assumptions assure the existence of a refining subsequence with a limit point x_*. By (3.17) we know that in the case of strict differentiability:

$$\forall\, d \in \mathbb{R}^n: \quad Df(x_*, d) = d^T \nabla f(x_*)\ .$$

Theorem 6.9 now guarantees the existence of a positive spanning set $\{p_i\}_{i=1}^{2n}$ so that

$$\forall\, i \in \{1,\dots,2n\}: \quad p_i^T \nabla f(x_*) \ge 0\ .$$

By Definition 4.1 we find

$$\forall\, d \in \mathbb{R}^n \;\; \exists\, \lambda_1, \ldots, \lambda_{2n} \geq 0 : \quad d = \sum_{i=1}^{2n} \lambda_i p_i \;.$$

Finally, we can conclude $\forall\, d \in \mathbb{R}^n$

$$\left.\begin{array}{rcl} d^T \nabla f(x_*) & = & \sum\limits_{i=1}^{2n} \lambda_i p_i^T \nabla f(x_*) \geq 0 \\[2ex] -d^T \nabla f(x_*) & = & \sum\limits_{i=1}^{2n} \tilde{\lambda}_i p_i^T \nabla f(x_*) \geq 0 \end{array}\right\} \Rightarrow \quad \nabla f(x_*) = O \;.$$

■

The final conclusion of this proof takes advantage of the linearity that is given by the strict differentiability. However, remembering our strategy of Chapter 5 the sublinearity of the Clarke derivative should be sufficient to proof the necessary condition we introduced in Chapter 3.

6.2.3 Analysis based on the Clarke derivative

At first we present another version of Theorem 6.10 that assures the necessary condition (3.11). We can do this by only assuming Lipschitz continuity of f around x_*. This is not just a relaxation of the requirement of strict differentiability. Actually it is the first convergence theorem in the context of GPS methods free of any classical differentiability assumption. What we loose by this generalization has been analyzed in Section 3.4.

Theorem 6.11.
Let $x_0 \in \mathbb{R}^n$ be the starting point for a GPS method and $f : \mathbb{R}^n \to \mathbb{R}$ be such that $L(x_0)$ is bounded. Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of iterates produced by the GPS method with objective f and let x_* be the limit of a refining subsequence. If f is Lipschitz continuous around x_*, then

$$\forall\, d \in \mathbb{R}^n : \quad Df(x_*, d) \geq 0 \;.$$

Proof:
This proof works out analogously to the one before. At first Theorem 6.9 guarantees the existence of a positive spanning set $\{p_i\}_{i=1}^{2n}$ such that

$$\forall\, i \in \{1, \ldots, 2n\} : \quad Df(x_*, p_i) \geq 0 \;.$$

By Definition 4.1 we can find for any vector $d \in \mathbb{R}^n$ nonnegative constants $\lambda_1, \dots, \lambda_{2n}$ such that

$$d = \sum_{i=1}^{2n} \lambda_i p_i \ .$$

Now by the sublinearity of $Df(x_*, \cdot)$ we can conclude

$$\forall\, d \in \mathbb{R}^n : \quad Df(x_*, d) = Df\Big(x_*, \sum_{i=1}^{2n} \lambda_i p_i\Big) \overset{(2.1)}{\geq} \sum_{i=1}^{2n} \lambda_i Df(x_*, p_i) \geq 0 \ .$$

■

But this is not all we can achieve. We are even able to allow inexact function evaluations by taking the set-valued approach.

We assume to be in a situation where we want to minimize a function $f : \mathbb{R}^n \to \mathbb{R}$. However, we can only compute function values f_c that are elements of the image set of an envelope mapping F_f around f given by (3.1), compare Section 5.3. If the inexactness is not too big, we can preserve our convergence result of Theorem 6.11. To be more precise assume that $\{p_i\}_{i=1}^{2n}$ is a positive spanning set associated with a refining subsequence $\{x_j\}_{j\in J}$ and step size parameters $\{h_j\}_{j\in J}$. Let for $j \in J$ denote $B_j(x_j)$ the smallest ball around x_j containing the set of vectors $\{x_j + p_i\}_{i=1}^{2n}$. We then have to require for the lower and upper bounds f_-, f_+ of the envelope mapping, compare with Equation (5.9),

$$\lim_{\substack{j\to\infty \\ j\in J}} \frac{\|f_+ - f_-\|_{B_j(x_j)}}{h_j} \quad = \quad 0 \ . \tag{6.14}$$

Theorem 6.12.
Let $x_0 \in \mathbb{R}^n$ be the starting point for a GPS algorithm and $f : \mathbb{R}^n \to \mathbb{R}$. Let $f_-, f_+ : \mathbb{R}^n \to \mathbb{R}$ and F_f be the envelope mapping around f given by (3.1). Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of iterates produced by the GPS method with objective function values $f_c(\cdot) \in F_f(\cdot)$. Let $L_c(x_0) := \{x \in \mathbb{R}^n \,|\, f_c(x) \leq f_c(x_0)\}$ be bounded and x_* be the limit of a refining subsequence with associated positive spanning set $\{p_i\}_{i=1}^{2n}$. If f is Lipschitz continuous around x_* and (6.14) holds, then

$$\forall\, d \in \mathbb{R}^n : \quad Df(x_*, d) \geq 0 \ .$$

Proof:
In order to apply Lemma 5.3, we especially have to show

$$\forall\, i \in \{1, \dots, 2n\} : \limsup_{\substack{j\to\infty \\ j\in J}} \frac{f_-(x_j + p_i) - f_-(x_j)}{h_j} \geq 0 \ .$$

The remaining prerequisites of the lemma are already contained in the assumptions of our theorem. For this purpose let $i \in \{1, \ldots, 2n\}$ be arbitrary, then

$$\limsup_{\substack{j\to\infty \\ j\in J}} \frac{f_-(x_j+p_i) - f_-(x_j)}{h_j} \overset{f_c \geq f_-}{\geq} \limsup_{\substack{j\to\infty \\ j\in J}} \frac{f_-(x_j+p_i) - f_c(x_j)}{h_j}$$

$$\overset{f_+ \geq f_c}{\geq} \limsup_{\substack{j\to\infty \\ j\in J}} \left[\frac{f_-(x_j+p_i) - f_+(x_j+p_i)}{h_j} + \frac{f_c(x_j+p_i) - f_c(x_j)}{h_j}\right]$$

$$\overset{(6.13)}{\geq} \limsup_{\substack{j\to\infty \\ j\in J}} \frac{f_-(x_j+p_i) - f_+(x_j+p_i)}{h_j} \overset{(6.14)}{\geq} 0 \ .$$

Thus, we can apply Lemma 5.3 to conclude

$$\forall\, d \in \mathrm{I\!R}^n : \quad Df(x_*, d) \geq 0 \ .$$

■

At this point we want to stop our theoretical discussion. The time has come to check the gained results in practice. In the following chapter both algorithms, Implicit Filtering and Pattern Search, are tested with a specially designed objective function.

Chapter 7

Numerical Computations

In the last two chapters we have analyzed the theoretical convergence issues of Implicit Filtering and Pattern Search. We now want to confirm them by some numerical computations. They also allow us to interpret the necessary theoretical assumptions and see how they can be used in practice.

In Section 7.1 we test an MATLAB implementation of the modified Implicit Filtering method of Section 5.3. After that we do the same with a simple Pattern Search method. The numerical results produced by the corresponding MATLAB-code are presented in Section 7.2. In both cases we use the same sample function f_c. It is the sum of a function f_L, which is locally Lipschitz continuous around the origin and a randomly generated additional term φ. The latter shall mimic the possible noise and converges to zero at the origin, where f_L attains its global minimum. In the symbols of Equation (5.8) the functions are for $x = \binom{x_1}{x_2} \in \mathbb{R}^2$ given by

$$
\begin{aligned}
f_L(x) \quad &:= \quad \begin{cases} \|x\| & \text{if } \|x\| < 1 \\ 1.5 + \sqrt{\|x\| - 1} & \text{if } \|x\| \geq 1 \quad , \ x_1 \cdot x_2 > 0 \\ 1 + \frac{|x_1| + |x_2|}{2} & \text{if } \|x\| \geq 1 \quad , \ x_1 \cdot x_2 \leq 0 \end{cases} \\
\varphi(x) \quad &\in \quad \Big[-\frac{f_L(x)}{10} \, , \, \frac{f_L(x)}{10} \Big] \\
f_c(x) \quad &:= \quad f_L(x) + \varphi(x) \, .
\end{aligned}
\tag{7.1}
$$

Note that $\min_{x \in \mathbb{R}^2} f_L(x) = f_L(O) = 0$ and f_L is really just Lipschitz continuous around the origin. It is not differentiable there. In addition we observe that this function is discontinuous on the unit circle and on the axes outside the unit circle. Figure 7.1 shows a graph of f_L and a possible realization of f_c .

It is easy to see that the noise decreases as we approach the origin. However, the

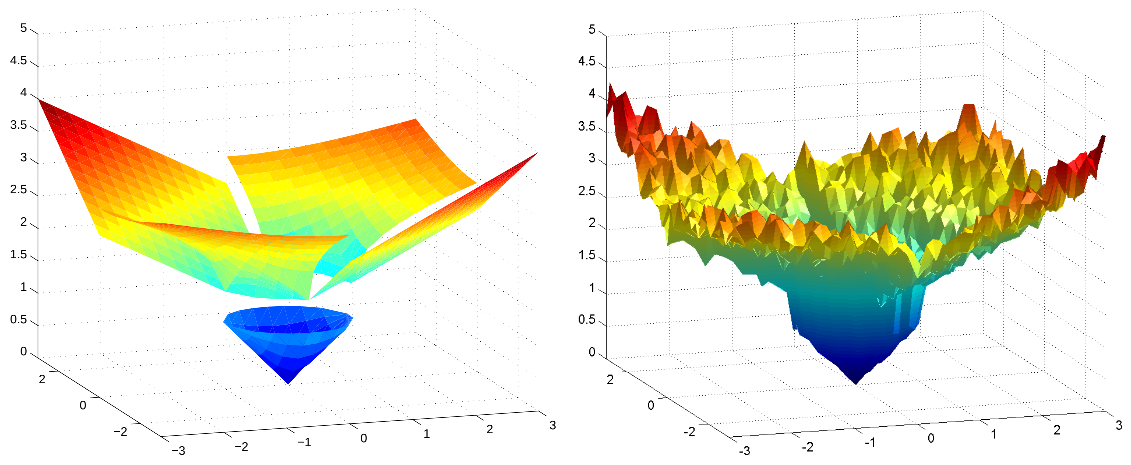

Figure 7.1: Graph of the underlying locally Lipschitzian objective function f_L (left) and some randomly perturbed function values f_c (right)

random perturbation has obviously a strong effect outside the unit circle. Both algorithms can deal with this problem, if the parameters are chosen carefully.

7.1 Modified Implicit Filtering

In this section we discuss the numerical results produced by an implementation of the modified Implicit Filtering method. The plain MATLAB-Code can be found in the appendix and an outline of the algorithm is provided in Section 5.3. There we also proved its theoretical convergence to a critical point under certain assumptions. Namely we had to require that

- f_L is Lipschitz continuous around a limit point of the sequence of iterates,
- there are at most finitely many line search failures and
- for the error φ, the sequence of scales $\{h_k\}_{k=1}^{\infty}$ and the simplices $\{S_k\}_{k=1}^{\infty}$ we had to insist on

$$\lim_{k\to\infty} \frac{\|\varphi\|_{S_k \cup R(S_k)}}{h_k} = 0 \; . \tag{7.2}$$

We already mentioned that the function f_L we use for our computations is Lipschitzian around its global minimizer. In Chapter 5 we learned that there is not much to do about the second assumption. Thus, we now focus on the third assumption and how it can determine the success of Implicit Filtering. Soon we learn how to adjust the parameters of the algorithm so that this last requirement is fulfilled. As a first guess we choose the following parameters for the optimization:

- $MAXH = 2^{-1}$ and $MINH = 2^{-8}$ as bounds for the scales h,
- $MAXIT = 6$ as maximum number of line search iterations,
- $MAXCUT = 8$ as maximum number of backtracking steps,
- $\alpha = 10^{-4}$, $\beta = 0.6$ and $\lambda = 1$ as parameters for the sufficient decrease condition,
- $\tau = 10^{-1}$ as tolerance for the termination criterion.

Since our objective function is defined in two different ways in the quadrants of the x_1, x_2-plane, we compare the different starting points $x_0^1 = \binom{-20}{20}$ and $x_0^2 = \binom{20}{20}$, see Figure 7.2. Obviously in both cases the algorithm fails to find the global minimum.

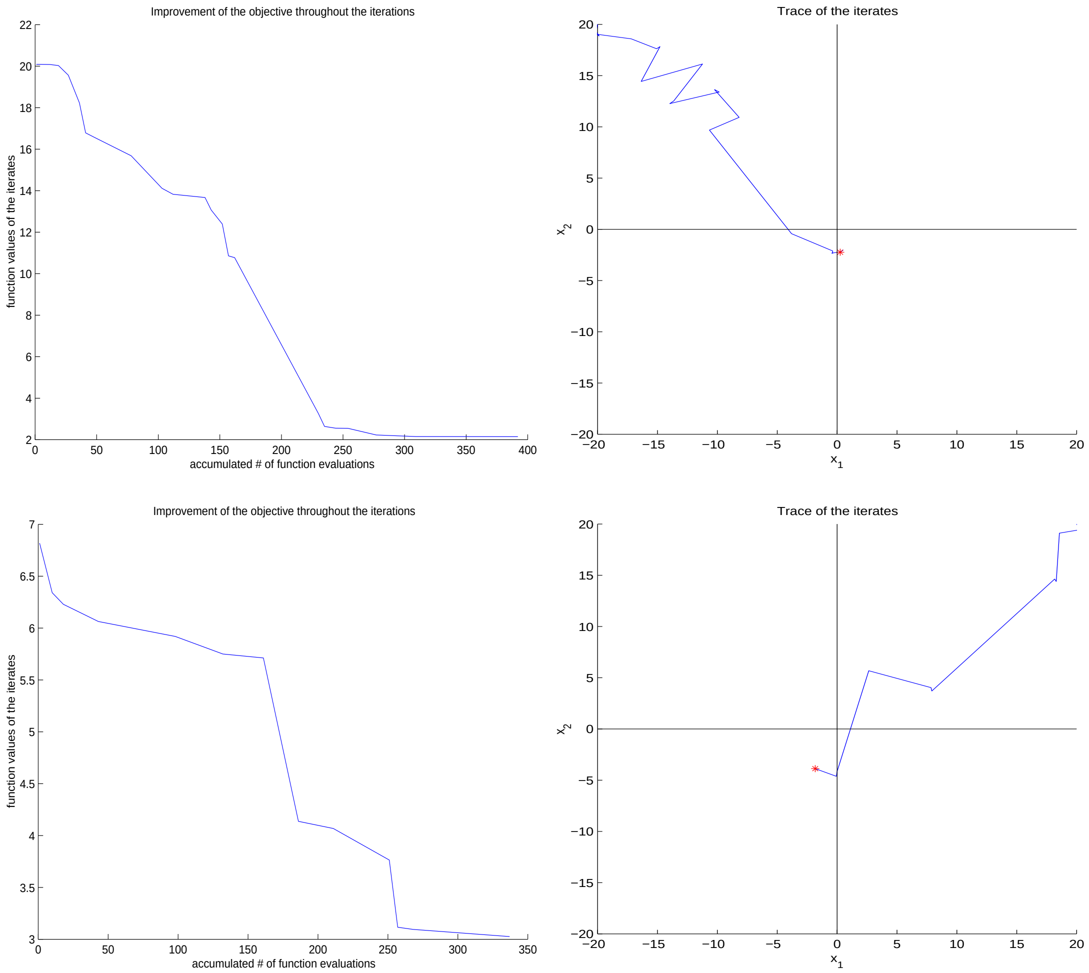

Figure 7.2: Results produced by the modified Implicit Filtering algorithm with a bad choice for the parameters and the starting iterates x_0^1 (top row) and x_0^2 (bottom row)

The traces of the iterates show that the search was conducted in the right direction. It looks like it just stopped too early. But if we look at the evolution of the function values, another problem can be noticed. Especially for the starting point x_0^1 we recognize that the function values do not change significantly throughout the last 150 function evaluations. The reason for this behavior is a violation of Equation (7.2). The ratio of the magnitude of the error and the scale does not approach zero. Thus the finite-difference approximations are not controlled by the behavior of f_L. Instead the noise φ has a major influence and misleads the algorithm. A lot of stencil failures are the consequence as one can easily detect by creating an additional output.

To avoid this failure, we might start with an augmented initial scale $MAXH$. This should help to reduce the effect of the noise. For basically the same reason we use a bigger initial step size λ for the line search. In order to maintain small step sizes for the final iterations, we increase the number of backtracking steps $MAXCUT$. Since we start with large scales, we have to take special care of the termination criterion. A smaller value of τ avoids an accumulation of false terminations. Altogether we made the following changes to get the results of Figure 7.3: $MAXH = 12$, $MAXCUT = 12$, $\lambda = 8$, $\tau = 0.005$.

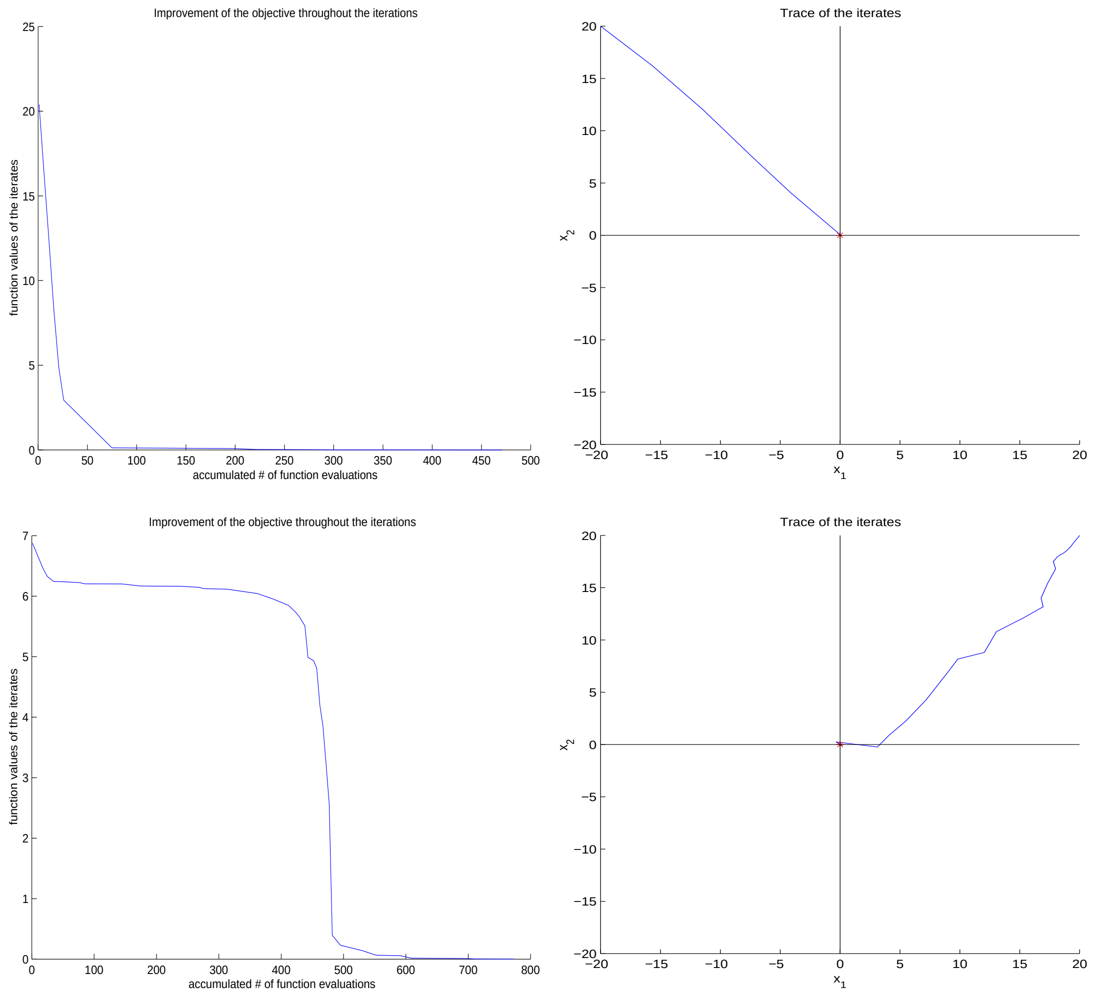

Figure 7.3: Results produced by the modified Implicit Filtering algorithm with carefully chosen parameters and the starting iterates x_0^1 (top row) and x_0^2 (bottom row)

Now the algorithm produces as a limit point a minimizer of our objective function. This happens for both initial iterates x_0^1 and x_0^2. But Figure 7.3 shows more than that. There seem to be different levels of difficulty in finding the minimizer. Obviously starting at x_0^2 causes more problems. This becomes apparent when we compare the development of the function values in Figure 7.3 (left column). Starting at x_0^1 we have immediately a fast progress, while starting at x_0^2 leads to a plateau of stagnation first. The reason for that behavior is the structure of the objective f_L . The starting points we picked are in different regions of the domain of f_L. For the initial iterate x_0^2 we first have to deal with the function $g(x) = 1.5 + \sqrt{\|x\| - 1}$. For $x = x_0^2 = \binom{20}{20}$ this function has a relatively small slope. Consequently in this region the error introduced by φ has a perceptible effect. We observe this as well from the trace of the iterates (right column of Figure 7.3). While for the initial iterate x_0^1 we proceed directly towards the origin, starting at x_0^2 results in a zigzag path. This unpredictable behavior is caused by the random influence of the error function. Also, if we restart the algorithm with the same parameters and the same initial iterate, the path of the iterates would change due to the randomly produced noise. It may even occur that the algorithm gets stuck on some plateau. But by changing the parameters once more in the same direction we did before, we can ensure the convergence towards the origin.

One might ask what happens if we enlarge the possible error. If we also adjust the parameters of the algorithm, it stays stable at first. This is possible up to a noise $\varphi(x)$ that makes up a quarter of the difference between the actual value $f_L(x)$ and the minimum value $f_L(O) = 0$. In other words we only have to require $\varphi(x)$ to be in $[-\frac{f_L(x)}{4} \, , \, \frac{f_L(x)}{4}]$. For the starting point x_0^1 we can allow even more noise. The algorithm still finds its way to the origin, if we allow an error with a magnitude of up to $\frac{1}{2}(f_L(x) - min_{x \in \mathbb{R}^2} f_L(x))$. But we have to make the right choice for the parameters. Figure 7.4 shows a landscape of computed function values and a trace

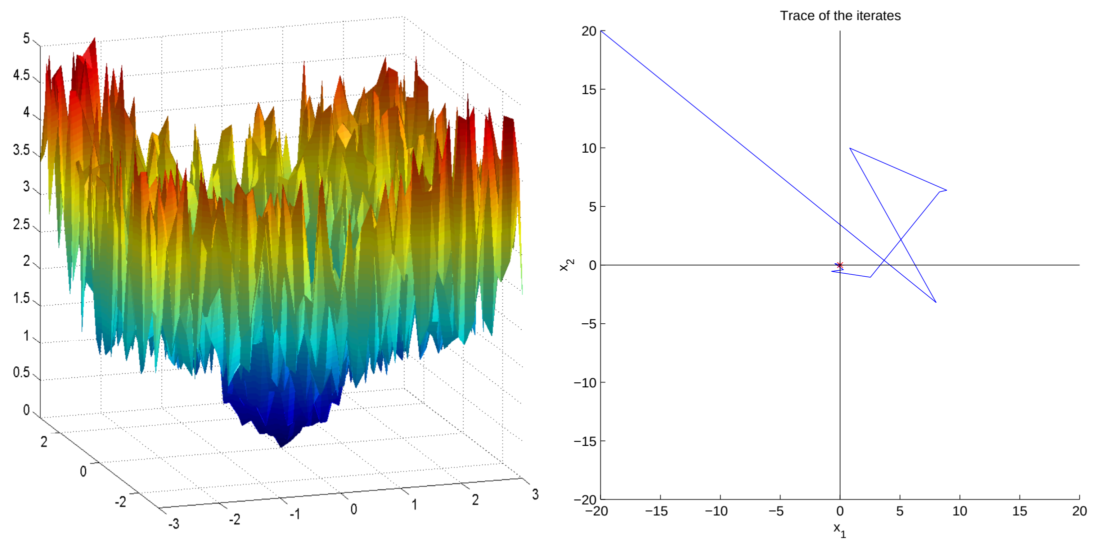

Figure 7.4: Extremely noisy function and trace of iterates produced by the modified Implicit Filtering algorithm with starting iterate x_0^1

of iterates for this extreme case. The effect of the noise is clearly visible in the trace of the iterates. At first they are jumping back and forth, but finally they converge towards the minimizer.

However, the adjustments we have to make slow down the algorithm, because more computations are needed. This can already be observed by comparing the total number of function evaluations of Figure 7.2 and 7.3. A similar phenomenon can be expected for the Pattern Search method. In the next section we point out that the choice of the parameters is even more crucial for our GPS-method. Pattern Search methods in general do not use specific information about the function to generate the trial steps. Thus, it becomes more important to adapt the parameters according to the problem to solve.

7.2 A Pattern Search Method

We now discuss the numerical results produced by a self-programmed Pattern Search method. It is simple in the way that it only uses a minimum of parameters. For example, we only use one factor ω_1 for the expansion of the step size h and our pattern just consists of the necessary positive spanning set and no additional vectors. We also use only one positive spanning set. The update of the pattern is nothing more but a permutation of its columns such that the last successful direction is investigated first in the next iteration. This may reduce the number of function evaluations and thus accelerate the algorithm. The corresponding MATLAB-code can be found in the appendix. For further information we refer to Chapter 6. The assumptions needed for the proofs of that chapter help to understand the numerical results presented now. We recall that

- f_L has to be Lipschitz continuous around the limit point of a refining subsequence of iterates $\{x_j\}_{j\in J}$,
- $L_c(x_0) = \{x \in \mathrm{I\!R}^2 \,|\, f_c(x) \le f_c(x_0)\}$ has to be bounded and
- for the error φ, the sequence of step size parameters $\{h_j\}_{j\in J}$ and some balls $\{B_j(x_j)\}_{j\in J}$ around the iterates we had to insist on

$$\lim_{\substack{j\to\infty \\ j\in J}} \frac{\|\varphi\|_{B_j(x_j)}}{h_j} = 0\ . \tag{7.3}$$

The first assumption is satisfied because of (7.1). Since we choose the same starting points x_0^1, x_0^2 as for the Implicit Filtering method in the previous section, the boundedness of $L_c(x_0)$ is assured as well. Thus, again everything boils down to the fulfillment of (7.3). In this section we explain how the algorithm has to be adjusted according to this observation. For the first optimization runs, we choose the following parameters:

- $\mathcal{M} = \{$all nonsingular combinations of $\pm e_1, \pm e_2\}$ as finite set of nonsingular integer matrices and $B = I_2$ as basis matrix,
- $h_0 = 1$ as initial stepsize as well as $\tau = 1.5$ and $-\omega_0 = \omega_1 = 1$ as parameters for the step size update,
- $MAXIT = 60$ as maximum number of Pattern Search iterations.

The results, which are displayed in Figure 7.5, do not look very promising. In fact the algorithm gets trapped immediately in one of the local minima created by the noise. The reason for that failure is our bad choice of parameters. At first our initial step size is too small to filter the noise. Second the step size reduction of $\tau^{\omega_0} = \frac{1}{1.5}$ is too big. After only a few unsuccessful iterations the step size is so small that the algorithm gets stuck in one of the fake local minima. Also the very limited set of only 4 search directions is probably not the best choice. Especially, because it is not designed for this particular objective function.

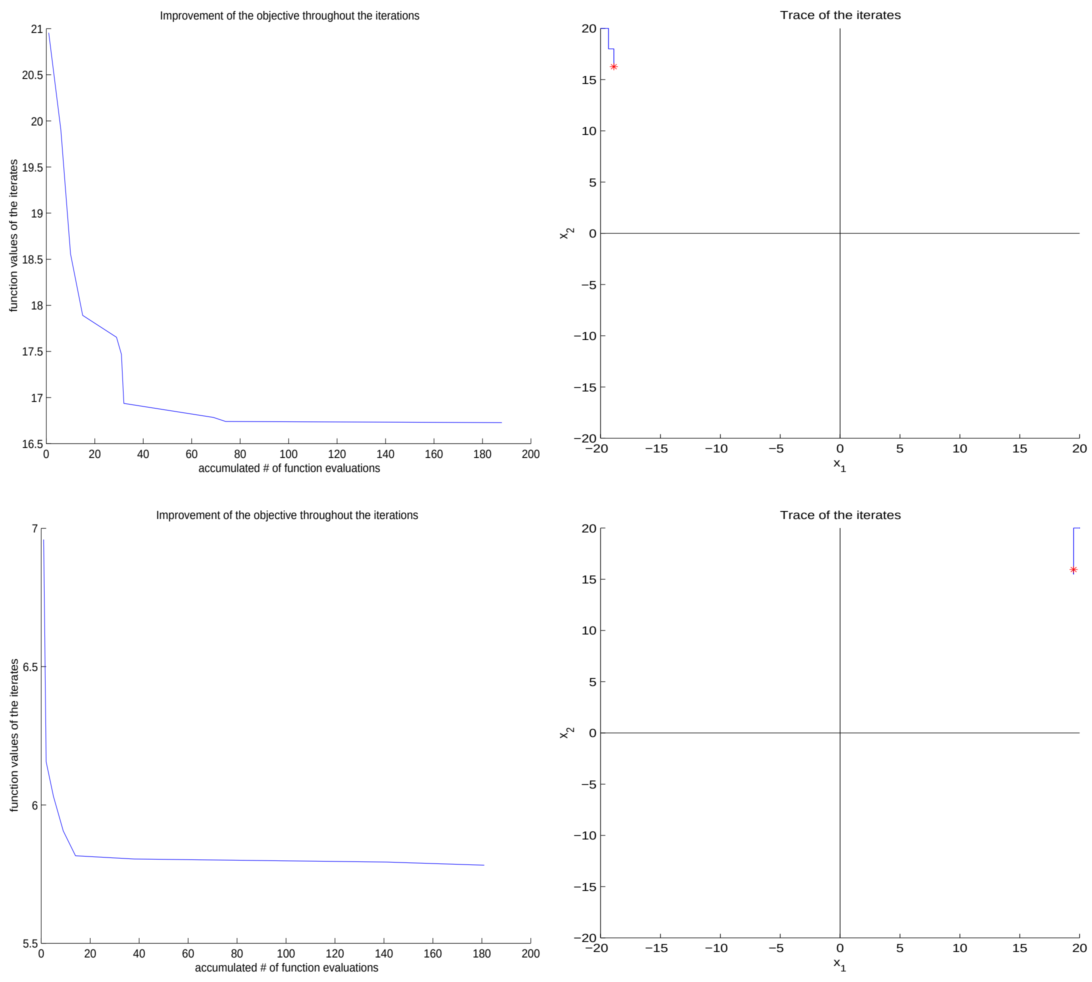

Figure 7.5: Results produced by our Pattern Search implementation with starting iterates x_0^1 (top row) and x_0^2 (bottom row) and badly chosen parameters

Despite these facts the evolution of the function values (left column in Fig. 7.5) looks like a typical successful Pattern Search output as we detect later. The fast progress of the initial phase slows down throughout the optimization process. However, since we know what our objective looks like, we are well aware of how far away we are from the actual minimum. This is confirmed by the trace of iterates (right column in Fig. 7.5). To improve the performance of our algorithm, we have to change just a single parameter. If we augment the initial step size parameter h_0 from 1 to 10, the optimization routine finds the minimum located at the origin. The reason for this success as well as for the failure before can be revealed by looking at Equation (7.3). A bigger initial step size causes bigger steps throughout the iterations. This helps to prevent the fraction in (7.3) from heading towards infinity. In addition promise these larger steps a faster progress towards the minimizer. Figure 7.6 shows the corresponding numerical results. Like for Implicit Filtering we detect that the stronger influence of the error in the region surrounding the starting iterate x_0^2 slows down the Pattern Search method.

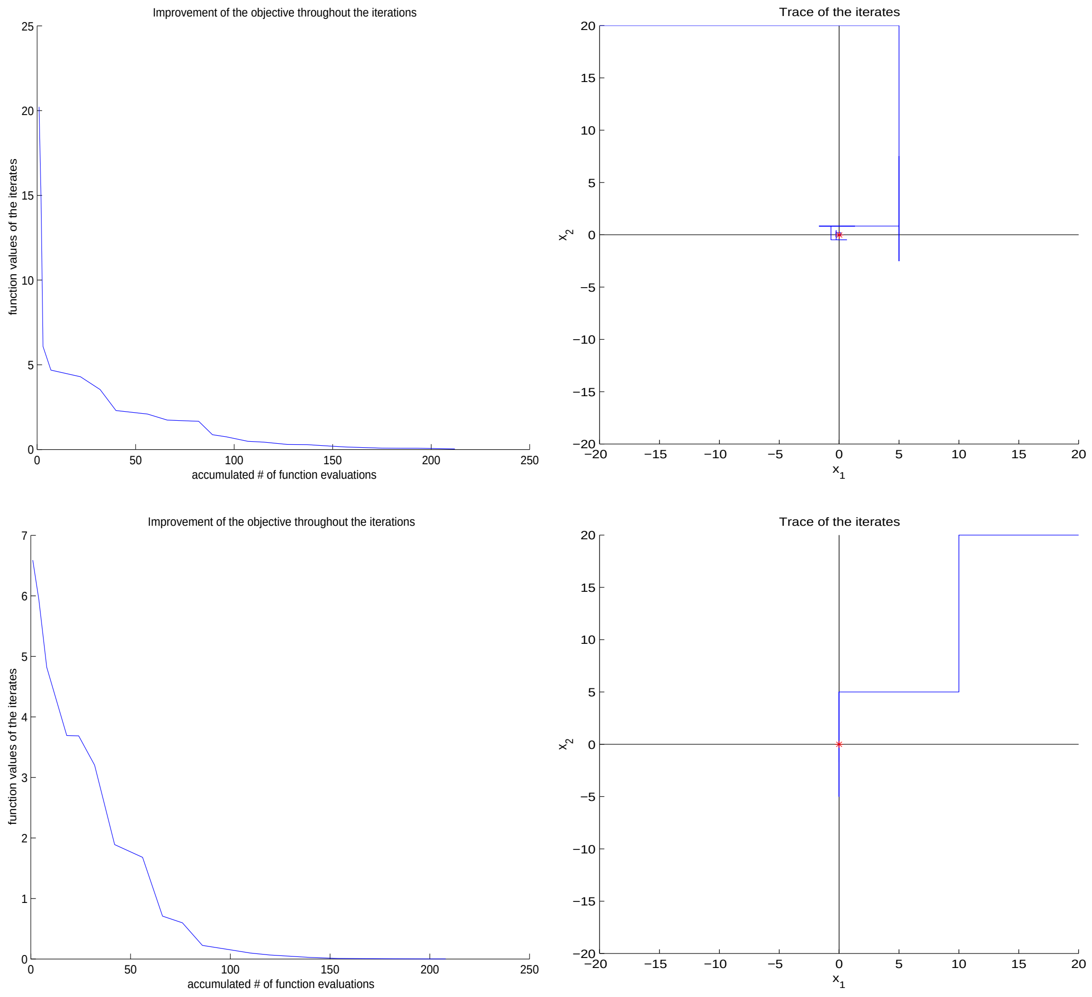

Figure 7.6: Results produced by the Pattern Search implementation with the same starting iterates but a bigger initial step size

Naturally, the same question arises that we already asked in Section 7.1. How much noise can the algorithm handle, if we choose the parameters carefully? Similar to Implicit Filtering there seems to be a limit for the success of our Pattern Search implementation. It is reached when the magnitude of the noise $\varphi(x)$ is about $\frac{1}{2}(f_L(x) - f_L(O))$. To get the results of Figure 7.7, we have to adjust several parameters. First of all, we increase the initial step size to $h_0 = 20$. In addition we lower the step size factor to $\tau = 1.1$. This slower adaption of the step size reduces the risk of getting stuck in a local minimum. But on the other side this change makes more iterations necessary. Thus we allow the algorithm to do more computations by shifting $MAXIT$ to 200. Once more we observe that an increasing level of noise slows down the algorithm. Another way to enhance the performance of the algorithm would be to use more

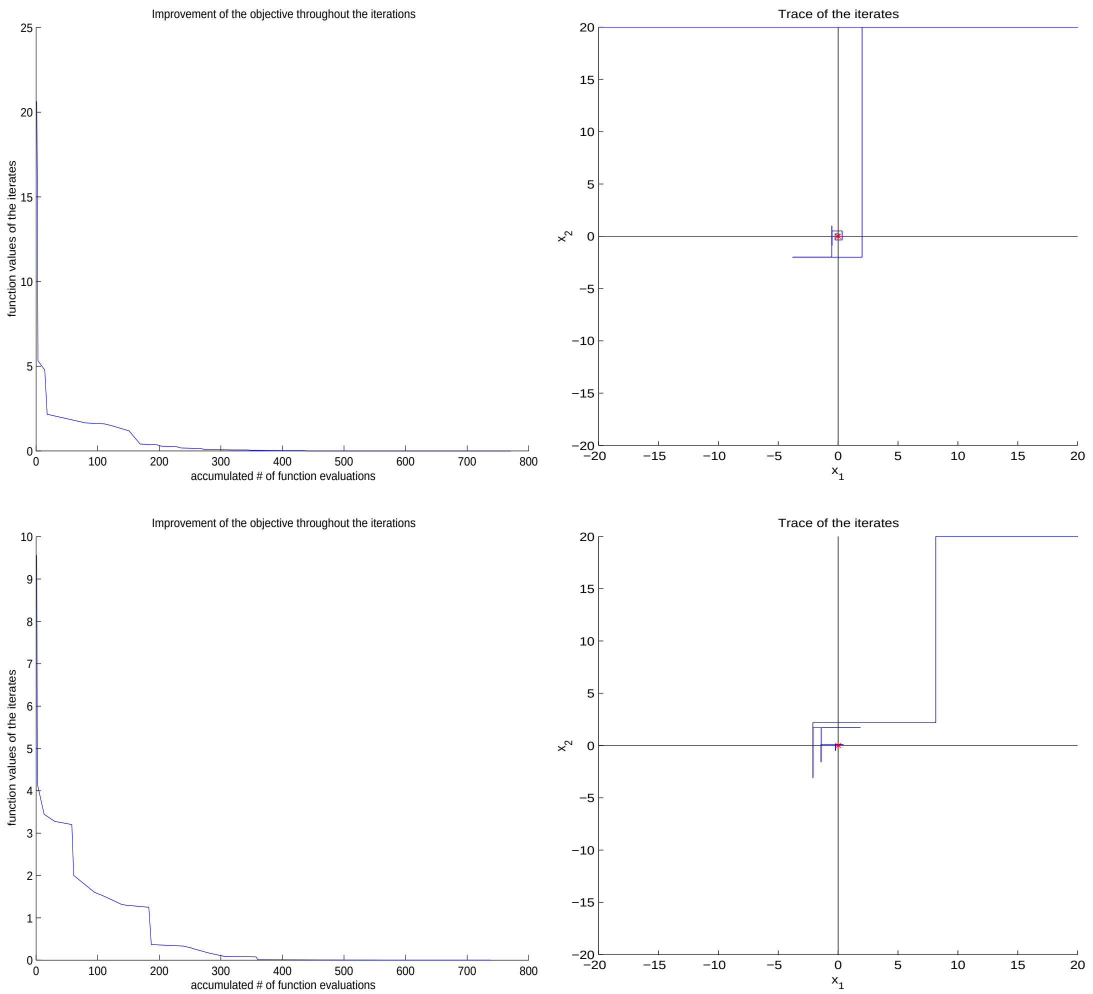

Figure 7.7: Results produced by the Pattern Search implementation with an extreme level of noise and carefully chosen parameters for our standard starting iterates

search directions than just one positive spanning set. This can extend the limits for the admissible noise a little more, but only at the expense of a slower algorithm. In general it depends on the specifics of the application, how far one wants or needs to decelerate the Pattern Search method.

This chapter has shown that both methods, Implicit Filtering and Pattern Search, can handle the optimization of a noisy and non-smooth function. But it is important to adjust the parameters of the algorithms according to the problem. We have also learned how the necessary theoretical assumptions of the Chapters 5 and 6 affect the practical performance of the routines. This new knowledge encourages testing the algorithms in further applications. Also the correspondence between theory and practice motivates extending the presented theory involving the Clarke derivative to the constrained case. There the algorithms are already in use and produce reasonable results.

Chapter 8

Conclusions and Outlook

In this work we developed a new tool for the convergence analysis of optimization routines. It is particularly designed for objective functions that are neither smooth nor can be evaluated exactly. Both phenomena are the subjects of active research and individually lead to interesting scientific discussions. In order to handle the non-smooth case, we introduced the Clarke derivative. Additionally, we used set-valued mappings to take care of the possible inaccuracy. This led to an alternative necessary condition for a minimizer.

We successfully applied this tool to prove new convergence results for the unconstrained versions of Implicit Filtering and General Pattern Search. We showed for a limit point of the sequence of iterates produced by either of the two methods that it satisfies our alternative necessary condition. This is in so far remarkable as we allowed the objective function to be both noisy and non-smooth at once. We only had to assume local Lipschitz continuity of the unperturbed function and some limiting behavior of the noise.

It was the latter that came in handy when we looked at practical applications of the algorithm. For Implicit Filtering we had to require that

$$\lim_{k\to\infty} \frac{\|f_+ - f_-\|_{S_k \cup R(S_k)}}{h_k} = 0 \, . \tag{8.1}$$

A similar condition had to be imposed for the Pattern Search methods. We found out that the key to a successful optimization run is a reasonable choice for the initial value h_0 and the parameters involved in its update. We were able to deduce this rule from our convergence theorems. In addition, the executed numerical computations confirmed this observation.

We also found out in practice that there are boundaries for the permissible inaccuracy. When we raised the noise above a certain level, the algorithms did not produce the actual minimizer as a limit point. However, the methods were able to move from the initial iterate towards the minimizer. They made some progress, but could not accomplish the whole process of finding a minimum.

It would be helpful to know how big the difference between computational output and the actual optimal value is at most. This question motivates further work. It might be possible to replace (8.1) by some weaker condition and then make some statement about the worst error that can happen.

Alternately we could focus on objective functions that are not even Lipschitz continuous. We only needed this property to obtain an expression of the Clarke derivative that we could handle numerically, namely

$$Df(x,d) = \limsup_{\substack{x_k \to x \\ h_k \downarrow 0}} \frac{f(x_k + h_k d) - f(x_k)}{h_k} \ .$$

This basically depended on the algorithms we examined. Probably we can find other optimization routines that can cope with more complicated expressions of the Clarke derivative. Apart from this it should in general be possible to apply our new analytical tool to some other optimization methods.

Last but not least we should be interested in developing this tool a little further. The results we presented for the unconstrained case in this text encourage future work towards a generalization for the constrained case. Even though it seems likely that efforts in this direction can be successful, there is no guarantee for that. Thus, we can only hope for the best, but have to expect the worst.

Bibliography

[1] C. Audet and J. E. Dennis, Jr. Pattern search algorithms for mixed variable programming. Technical Report TR99-02, Rice University, Department of Computational and Applied Mathematics, Houston, TX, 1999.

[2] C. Audet and J. E. Dennis, Jr. Analysis of generalized pattern searches. Technical Report TR00-07, Rice University, Department of Computational and Applied Mathematics, Houston, TX, October 2000.

[3] C. Audet and J. E. Dennis, Jr. A pattern search method for nonlinear programming without derivatives. Technical Report TR00-09, Rice University, Department of Computational and Applied Mathematics, Houston, TX, March 2000.

[4] A. Battermann. *Mathematical Optimization Methods for the Remediation of Ground Water Contaminations*. PhD thesis, University of Trier, Trier, Germany, February 2001.

[5] A. Battermann, T. Coffey, J. M. Gablonsky, K. R. Kavanagh, C. T. Kelley, C. T. Miller, and A. Patrick. Solution of a groundwater control problem with implicit filtering. Technical Report CRSC-TR00-30, North Carolina State University, Center for Research in Scientific Computation, Raleigh, NC, 2000. Submitted.

[6] G. L. Bilbro, P. Gilmore, C. T. Kelley, D. E. Stoneking, and R. J. Trew. Yield optimization using a GaAs process simulator coupled to a physical device model. In *Proc. IEEE/Cornell Conference on Advanced Concepts in high Speed Devices and Circuits*, pages 374–383, Piscataway, NJ, 1991. IEEE.

[7] G. L. Bilbro, P. Gilmore, C. T. Kelley, D. E. Stoneking, and R. J. Trew. Yield optimization using a GaAs process simulator coupled to a physical device model. *IEEE Transactions on Microwave Theory and Techniques*, 40:1353–1363, 1992.

[8] D. Bortz and C. T. Kelley. The simplex gradient and noisy optimization problems. In J. Borggaard, J. Burns, E. Cliff, and S. Schreck, editors, *Computational Methods for Optimal Design and Control*, volume 24, pages 77–90, Boston, 1998. Birkhäuser.

[9] F. H. Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205(2):247–262, April 1975.

[10] F. H. Clarke. *Methods of Dynamic and Nonsmooth Optimization.* Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[11] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, N.J., 1983.

[12] J. E. Dennis, Jr. and V. Torczon. Direct search methods on parallel machines. *SIAM Journal of Optimization*, 1:448–474, 1991.

[13] P. Gilmore and C. T. Kelley. An implicit filtering algorithm for optimization of functions with many local minima. *SIAM Journal of Optimization*, 5:269–285, 1995.

[14] P. Gilmore, C. T. Kelley, R. J. Trew, and T. A. Winslow. Doping profiles for optimum class b performance of GaAs MESFET amplifiers. In *Proc. IEEE/Cornell Conference on Advanced Concepts in high Speed Devices and Circuits*, pages 188–197, Piscataway, NJ, 1991. IEEE.

[15] P. Gilmore, C. T. Kelley, R. J. Trew, and T. A. Winslow. Simulated performance optimization GaAs MESFET amplifiers. In *Proc. IEEE/Cornell Conference on Advanced Concepts in high Speed Devices and Circuits*, pages 393–402, Piscataway, NJ, 1991. IEEE.

[16] A. Götz and J. Jahn. The Lagrangian multiplier rule in set-valued optimization. *SIAM Journal of Optimization*, 10(2):331–344, 1999.

[17] R. Hooke and T. A. Jeeves. “Direct search” solution of numerical and statistical problems. *Journal Assoc. Comput. Mach.*, 8:212–229, 1961.

[18] J. Jahn and R. Rauh. Contingent epiderivatives and set-valued optimization. *Mathematical Methods in Operations Research*, 46:193–211, 1997.

[19] C. T. Kelley. *Iterative Methods for Optimization*, volume 18 of *Frontiers in applied mathematics.* SIAM, Philadelphia, 1999.

[20] R. M. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941, 2000.

[21] D. T. Luc. Contingent derivatives for set-valued maps and applications to vector optimization. *Math. Programming*, 50:99–111, 1991.

[22] L. Nachbin, editor. *Mathematical Analysis and Applications*, volume 7A of *Advances in Mathematics*, chapter 5 by J. P. Aubin, pages 159–229. Academic Press, New York, 1981.

[23] R. T. Rockafellar. *Convex analysis*, volume 28 of *Princeton Mathematical Series.* Princeton Univerity Press, Princeton, N.J., 1970.

[24] R. T. Rockafellar. Generalized directional derivatives and subgradients of non-convex functions. *Canadian Journal of Mathematics*, 32:157–180, 1980.

[25] R. T. Rockafellar. *The theory of subgradients and its applications to problems of optimization*, volume 1 of *Research and education in mathematics*. Heldermann, Berlin, 1981.

[26] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal of Optimization*, 7(1):1–25, February 1997.

Appendix A

MATLAB-Files

As announced in Chapter 7 we now list the computer codes used in this thesis. Both were programed by the author himself. However, without the previous work of other mathematicians this would not have been possible. Especially, the articles of V. Torczon were helpful to gain an insight into Pattern Search methods. We also want to mention C. T. Kelley and his publications on Implicit Filtering. In addition he has downloadable MATLAB-files for this optimization method on his webpage (`http://www4.ncsu.edu/~ctk/matlab_darts.html`). They are the basis on which our modified version was built.

The listings of this appendix only display the executed numerical computations. To create the output of Chapter 7, additional lines have to be fitted into the codes. The listed programs are also not totally documented. However, the purpose and use of all variables should be clear after the expositions of Chapter 5 and 6. Their actual values on the next pages represent the successful tuning for the extremely noisy test-function of Chapter 7. The MATLAB-file for simulating this perturbed locally Lipschitzian objective is also listed at the end of this appendix.

Modified Implicit Filtering Version

```
function xend = modimfil(f,x_0);

n      = length(x_0); % Dimension of the input vector
maxh   = 20;          % Size of the first scale h
minh   = 2^(-8);      % Lower bound for the scales
maxit  = 3*n;         % Number of line search iterations
maxcut = 30;          % Number of line search backtracking steps
alpha  = 10^(-3);     % Fraction for sufficient decrease (in (0,1))
lambda = 40;          % Initial step size for the line search
beta   = .75;         % Reduction parameter for the step size
tau    = .001;        % Error tolerance for the termination criterion
h      = maxh;        % First scale for the finite-diff. approx.
V      = eye(n);      % Matrix of simplex directions
x      = x_0;         % Starting point for the algorithm
f_x    = feval(f,x);  % Function value at current iterate

% Main Implicit Filtering loop
while h > minh;
   % Loop for Line Searches
   for m = 1:maxit;

      % necessary function evaluations
      for i = 1:n;
         f_for(i,1) =feval(f,x+h.*V(:,i));
         f_back(i,1)=feval(f,x-h.*V(:,i));
      end

      % test for stencil failure
      min_for  = min(f_for);
      min_back = min(f_back);
      if f_x < min(min_for, min_back);
         break;
      end

      % Calculation of the simplex gradients and their norms
      D_plus     = (f_for - f_x*ones(n,1))/h;
      D_minus    = (f_back - f_x*ones(n,1))/(-h);
      D_central  = (D_plus + D_minus)/2;
      norm_plus  = norm(D_plus);
      norm_minus = norm(D_minus);
      % test for termination criterion
      if .5*(norm_plus + norm_minus) <= tau*h;
         break;
      end

      % Line Search backtracking loop
      sufficient = 0;
      for l=0:maxcut;
         step = lambda*(beta^l);
         test = feval(f,x - step*D_central)-f_x;
         if test < -alpha*step*(norm_plus^2 + norm_minus^2)/2;
            sufficient = 1;
            x = x - step*D_central; % update the iterate x
            f_x = f_x+test;         % update function value
            break;
         end
      end

      % Test for Line Search failure
      if sufficient == 0;
         break;
      end

   end
   h = .9*h; % reduction of the scale
end

xend = x;
```

Simple Pattern Search Version

```
function xend = simplegps(f,x_0);

n      = length(x_0);    % Dimension of the input vector
maxit  = 200;            % Maximum number of GPS iterations
M      = sparse(eye(n)); % The first used nonsingular integer matrix
B      = sparse(eye(n)); % The basis matrix
tau    = 1.1;            % The basic step length factor
omega0 = -1;             % The exponent for step size reduction
omega1 = 1;              % The exponent for step size prolongation
x      = x_0;            % Starting point for the algorithm
h      = 15;             % Initial step size parameter
G      = [M -M];         % Initial positive spanning set
P      = B*G;            % Initial pattern matrix
f_x    = feval(f,x);     % Function value at current iterate

% Main loop for the Pattern Search
for k = 0:maxit;
   % Performing the exploratory moves
   for i = 1:2*n;
      f_trial = feval(f,x+h*P(:,i));
      if f_trial < f_x;
         x   = x+h*P(:,i);   % update the iterate x
         f_x = f_trial;      % update function value
         h   = h*tau^omega1; % update the step size h
         % Permute the matrix G so that in the next iteration
         % the last successful direction is searched first
         d1     = [1;zeros(2*n-1,1)];
         d2     = [0;ones(i-1,1);zeros(2*n-i,1)];
         d3      = [zeros(i,1);ones(2*n-i,1)];
         PERMUT = spdiags([d1 d2 d3],[-(i-1) 1 0],2*n,2*n);
         G      = G*PERMUT;
         P      = B*G;       % update pattern matrix
         break;
      elseif i == 2*n
         h = h*tau^omega0;   % update the step size h
      end
   end
end

xend = x;
```

Locally Lipschitz continuous objective function with high-level noise

```
function z=err(x);

% 2 variable test function
% for the numerical computations.

if norm(x) <= 1
   z = norm(x);
else
   if x(1)*x(2) > 0
     z = 1.5 + sqrt(norm(x)-1);
   else
     z = 1 + (abs(x(1))+abs(x(2)))/2;
   end
end

z = z*(1 + (rand-.5));
```

Vita

Name:	Andreas Krahnke
Date of birth:	July, 26th 1976
Place of birth:	Mayen, Germany
Nationality:	German

Education:

08/2000 - 05/2001	Master of Science in Mathematics, Virginia Polytechnic Institute & State University, Blacksburg, Virginia (U.S.A.) Graduating Spring 2001
10/1999 -	Hauptstudium in Wirtschaftmathematik, University of Trier (Germany) expected year of completion 2002
10/1997 - 10/1999	Prediploma in Wirtschaftmathematik, University of Trier (Germany)
07/1996 - 05/1997	Military service, Rheine, Coesfeld und Mendig (Germany)
08/1987 - 06/1996	Abitur, Kurfürst-Balduin-Gymnasium, Münstermaifeld (Germany)

Expierence:

08/2000 - 05/2001	Graduate Teaching Assistant, Virginia Polytechnic Institute & State University, Blacksburg, Virginia (U.S.A.)
10/1999 - 06/2000	Graduate Research Assistant, University of Trier (Germany), mentored by Dr. Ekkehard W. Sachs and Astrid Battermann, work on the Project "Mathematische Optimierungsmethoden bei der Sanierung von Grundwasserkontaminationen" funded through Stiftung Rheinland-Pfalz für Innovation