UNIVERSITÄT PADERBORN

Die Universität der Informationsgesellschaft

Minimization of Lipschitzian Piecewise Smooth Objective Functions

Der Fakultät für Elektrotechnik, Informatik und Mathematik

der Universität Paderborn

zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN

– Dr. rer. nat. –

vorgelegte Dissertation

von

Dipl.-Math. Sabrina Fiege

Paderborn, 2017

Gutachter: Prof. Dr. Andrea Walther
Prof. Dr. Andreas Griewank
Prof. Dr. Marc Steinbach

Tag der mündlichen Prüfung: 4. September 2017

Acknowledgments

Foremost, I would like to thank my advisor Prof. Dr. Andrea Walther for her helpfulness, support, and guidance over the years. I am grateful for the opportunity of studying and researching in the magnificent field of nonsmooth optimization. Her enthusiasm for applied mathematics was contagious and motivational for me.

Furthermore, I would like to thank Prof. Dr. Andreas Griewank for many fruitful and inspiring discussion as well as for his ongoing advice and insightful comments. In particular, I would like to thank him and Prof. Dr. Marc Steinbach for examining this doctoral thesis.

I am also very grateful to all my colleagues I worked with throughout the years. Especially, I would like to thank the members of the group of Mathematics and its Applications: Mladen Banovic, Olga Ebel, Benjamin Jurgelucks, Kshitij Kulshreshtha, Veronika Schulze, Maria Schütte, Karin Senske, and Tobias Steinle, for many long conversations, for their listenings at any time and for their companionship.

Last but certainly not least, I would like to thank friends and family for all their love and support. Above all I would like to thank my parents for their constant and cordial support that accompanied me through many years of study.

Thank you very much!

Abstract

Nonsmoothness is a typical characteristic of numerous optimization problems originating from both real world and scientific applications. Well known examples from practical optimization are minimax problems frequently used in robust optimization and the reformulation of a constrained optimization problem by adding ℓ_1- or ℓ_∞-penalty terms of constraint violations to the original function.

Although there are plenty of publications dealing with nonsmooth analysis and optimization, there are only a few state-of-the-art software tools available for nonsmooth optimization problems. Therefore, the purpose of this thesis is to develop, implement, and examine an algorithm for unconstrained, nonconvex, and nonsmooth optimization problems. It will be assumed that all nondifferentiabilities occurring in the objective function are caused by the absolute value function and those functions that can be expressed in terms of the absolute value function as the maximum and minimum function. Functions of this form will be called composite piecewise differentiable functions.

The idea of the optimization method LiPsMin developed in the scope of this thesis is the minimization of composite piecewise differentiable objective functions via successive piecewise linearization overestimated by a quadratic term. The minimization of the resulting local quadratic subproblem benefits from additional information obtained by exploiting the structure of the underlying piecewise linearization. Convergence results of LiPsMin towards first order optimal points are developed and the numerical performance of the algorithm is investigated by comparing it with other state-of-the-art nonsmooth optimization software packages.

Keywords: Nonsmooth optimization, Piecewise linearization, Algorithmic Differentiation

Zusammenfassung

Nichtglattheit ist eine typische Eigenschaft vieler Optimierungsprobleme, die ihren Ursprung sowohl in industriellen als auch in akademischen Anwendungen haben. Bekannte Beispiele sind unter anderem Minimax-Probleme aus der Robusten Optimierung sowie die Umformulierung beschränkter Optimierungsprobleme in unbeschränkte Probleme indem man die Beschränkungen als ℓ_1- oder ℓ_∞- Strafterme additiv zur Zielfunktion hinzufügt.

Obwohl es eine Vielzahl von Veröffentlichungen zu nichtglatter Analysis und Optimierung gibt, sind nur wenige moderne Software-Pakete für nicht-glatten Optimierungsprobleme verfügbar. Aus diesem Grund ist das Ziel dieser Dissertation die Entwicklung und Implementierung eines Algorithmus zur Lösung unbeschränkter, nichtkonvexer und nichtglatter Optimierungsprobleme. Darüber hinaus soll der vorgestellte Algorithmus im Rahmen dieses Promotionsprojekts getestet werden. Es wird angenommen, dass alle Nichtdifferenzierbarkeiten der Zielfunktion durch den Absolutbetrag verursacht werden. Dies umfasst auch Funktionen, deren Nichtdifferenzierbarkeiten mittels Absolutbetrag ausgedrückt werden können, wie z.B. die Minimum- und Maximumsfunktion. Funktionen dieser Form werden zusammengesetzte stückweise differenzierbare Funktionen genannt.

Die Idee des Optimierungsalgorithmus LiPsMin, der im Rahmen dieser Dissertation entwickelt wurde, ist die Minimierung einer zusammengesetzten stückweise differenzierbaren Funktionen durch wiederholtes Generieren lokaler Modelle der Zielfunktion. Diese Modelle setzen sich aus einer stückweisen Linearisierung und einem quadratische Term zusammen. Dabei profitiert die Minimierung des so entstandenen lokalen Modells von den zusätzlichen Informationen, die durch Strukturausnutzung der zu Grunde liegenden stückweisen Linearisierung gewonnen werden können. Die Untersuchung des Algorithmus LiPsMin wird durch Konvergenzergebnisse bzgl. optimaler Punkte erster Ordnung abgerundet. Abschließend wird die numerische Effizienz des Algorithmus untersucht, indem er mit anderen modernen Software-Paketen zur Lösung nichtglatter Optimierungsprobleme verglichen wird.

Stichworte: Nichtglatte Optimierung, Stückweise Linearisierung, Algorithmisches Differenzieren

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AD	Algorithmic differentiation
PL	Piecewise linearization
PS	Piecewise smooth
LiPsMin	Minimization routine for Lipschitzian piecewise smooth functions
PLMin	Minimization routine for piecewise linear functions

Notation

x	Variable $x \in \mathbb{R}^n$
d	Direction $d \in \mathbb{R}^n$
$B_\epsilon(x)$	Open ball with center x and radius $\epsilon > 0$
U	Subset of $\mathbb{R}^n$
$\text{relint}(U)$	Relative interior of U
$\text{aff}(U)$	Affine hull of U
$\text{conv}(U)$	Convex hull of U
$\text{cone}(U)$	Conic hull of U
U°	Polar cone of U
$K_U(x)$	Contingent cone to U at x
$N_U(x)$	Normal cone to U at x
$T_U(x)$	Tangent cone to U at x
$F_U(y)$	Max-Face of U corresponding to vector $y \in \mathbb{R}^n$
epi f	Epigraph of function f
$\mathcal{N}(f, \alpha)$	Level set of function f with $\alpha \in \mathbb{R}$
$I_f(x)$	Active index set of PS function f in x
$I_f^e(x)$	Essentially active index set of PS function f in x
$f'(x; d)$	Directional derivative of f at x in direction d

$f^\circ(x;d)$	Generalized directional derivative of f at x in direction d
$D_s f(x)$	Strict derivative of f at x
$\partial f(x)$	Subdifferential of f at x
$\partial_C f(x)$	Clarke's generalized derivative or Clarke's subdifferential of f at x
$\partial_L f(x)$	Limiting subdifferential of f at x
s	Number of absolute values occurring during function evaluation
z	Switching vector with $z \in \mathbb{R}^s$
σ	Signature vector with $\sigma \equiv \{-1, 0, 1\}^s \in \mathbb{R}^s$
$\mathcal{E}$	Set of essentially active signature vectors
P_σ	Polyhedron corresponding to signature σ
$\overline{P}_\sigma$	Closure of polyhedron P_σ
$\hat{P}_\sigma$	Extended Closure of polyhedron P_σ
f_σ	Selection function corresponding to signature σ
g_σ	Gradient of f_σ corresponding to signature σ with $\sigma \in \mathcal{E}$
$\Delta x, \Delta y, \Delta z$	Increments $\Delta x, \Delta y, \Delta z \in \mathbb{R}^n$ of variables x, y and z
$f_{PL,x}(\Delta x)$	PL of f for fixed x and argument Δx
$\Delta f(x;\Delta x)$	Increment function of $f_{PL,x}$ for fixed x and argument Δx
$\hat{f}_x(\Delta x)$	Quadratic model of f for fixed x and argument Δx
q	Quadratic penalty coefficient
$\breve{q}$	Overestimated quadratic penalty coefficient

1 Introduction

Nonsmoothness is a typical characteristic of numerous optimization problems originating from both real world and scientific applications. Typical examples of real world applications are phase changes in materials, certain minimal or maximal bounds on the usage of utilities in economies and recommender systems used by online retailers. Well known examples from practical optimization are minimax problems frequently used in robust optimization and the reformulation of a constrained optimization problem by adding ℓ_1- or ℓ_∞-penalty terms of constraint violations to the original function.

Development of Nonsmooth Theory and Optimization Methods

Nonsmooth optimization deals with objective functions and possibly constraint functions that are not necessarily everywhere differentiable. First optimization methods for convex problems were developed in the 1960s and 1970s as the *cutting-plane method* by J.R. Kelley [Kel60] and *subgradient methods* as well as the *gradient-type method with space-dilation* by N.Z. Shor [Sho79]. The required concepts such as subdifferentials and optimality conditions were introduced by T.R. Rockafellar in his fundamental book *Convex Analysis* [Roc70]. In the 1980s and later on more generalized classes of functions were considered, among others quasidifferential functions by V.F. Demyanov and L.V. Vasilev [DV85], Lipschitz continuous functions by F.H. Clarke [Cla83] and piecewise differentiable functions by S. Scholtes [Sch12].

In the 1970s first *bundle methods* for convex problems were introduced by C. Lemaréchal [Lem78] and K.C. Kiwiel [Kiw85]. Since then they represent an important class of nonsmooth optimization methods. A variety of extended bundle methods were developed as combinations with the *trust-region method* [Sch89, ANR16], with

Newton-type methods [LV98], and refinements as *proximal bundle methods* [LS97, MN92]. Additionally, bundle methods were adapted for nonconvex problems [Mif82], constrained problems [SS05], and multi-criteria problems [Mie98].

Simultaneously, the development of subgradient methods proceeded. Especially, *variable metric methods* are wide-spread such as BFGS methods by F.E. Curtis, T. Mitchell and M.L. Overton [CMO17], and a combination of the trust region algorithm and the *limited memory BFGS method* by G. Yuan, Z. Wei, and Z. Wang [YWW13].

At the beginning of the 2000s, J.V. Burke, A.S. Lewis and M.L. Overton presented an *gradient sampling algorithm* for nonconvex, nonsmooth problems in [BLO05] which they coupled later on with quasi-Newton type methods.

Since nonsmooth optimization is still a challenging and important field, this brief overview represents only a small extract of publications dealing with nonsmooth problems.

Idea and Purpose of this Thesis

Nevertheless, there are only a few state-of-the-art software tools available for nonsmooth optimization. Therefore, the purpose of this thesis is to develop, implement, and examine an algorithm for unconstrained, nonconvex, and nonsmooth optimization problems via successive piecewise linearization. This means in detail that optimization problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1.1}$$

will be considered where $f : \mathbb{R}^n \to \mathbb{R}$ is a Lipschitz continuous, piecewise smooth function. A piecewise smooth function in the sense of Scholtes is a function that is everywhere locally a continuous selection of finitely many continuously differentiable functions, see [Sch12]. Additionally, it will be assumed subsequently, that all nondifferentiabilities are caused by the absolute value function and those functions that can be expressed in terms of the absolute value function as the maximum and minimum function. Functions of this form will be called *composite piecewise differentiable functions* in the following. These assumptions are reasonable since the

absolute value function causes the nondifferentiabilities of numerous target functions arising from real world applications.

In the case of a convex, piecewise linear function the algorithm developed in this thesis generates with appropriate parameter settings the continuous steepest descent trajectory originally analyzed by J.-B. Hiriart-Urruty and C. Lemaréchal in the book *Convex Analysis and Minimization Algorithms I*, see [HUL93, Chap. 8, Sec 3.4]. In that book the non-convergence of the classical steepest-descent method with exact line search was considered and illustrated with the aid of the piecewise affine and even convex function

$$\begin{aligned} f : \mathbb{R}^2 \to \mathbb{R}, \quad & f(x) := \max\{f_0(x),\ f_{\pm 1}(x),\ f_{\pm 2}(x)\}, \\ \text{with} \quad & f_0(x) := -100,\ f_{\pm 1}(x) := 3x_1 \pm 2x_2,\ f_{\pm 2}(x) := 2x_1 \pm 5x_2. \end{aligned} \tag{1.2}$$

This function serves as a counter-example as can be seen in Fig. 1.1. Solely those points in which the function f_0 is active are minimal points. The initial point of the optimization run is $x_0 = (9, -3)$. The trajectory generated by the steepest descent method with exact line search zigzags. It converges towards the point $(0, 0)$ but it never attains this non-stationary point. The steepest descent direction towards the optimal points can not be found because of that and thus the method does not converge.

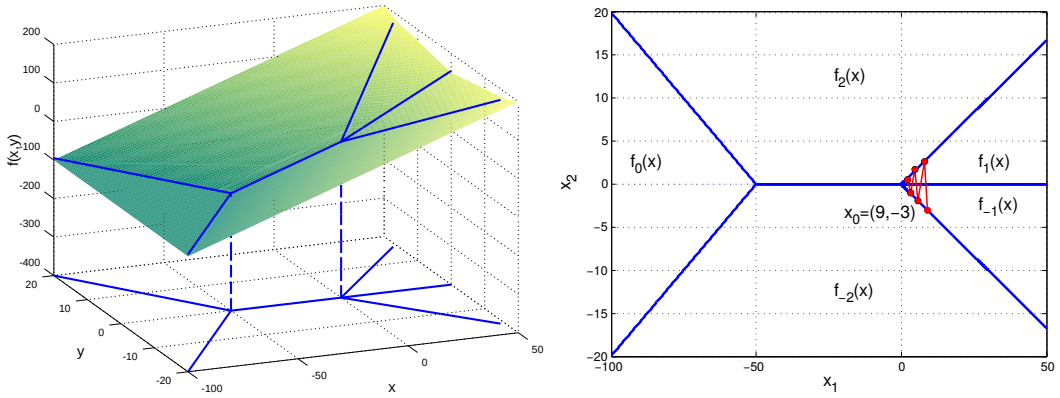


Figure 1.1: Left: Graph of function (1.2) and nondifferentiable points
Right: Zigzagging behavior of steepest descent method

Therefore, the method of the steepest descent trajectory is suggested for convex functions in [HUL93, Chap. 3.4]. The trajectory is described as the solution of the differential inclusion

$$\dot{x}(\tau) \in -\partial f(x(\tau)), \quad x(0) \text{ given.}$$

where $\tau \geq 0$. For convex functions, convergence of the method of the steepest descent trajectory is proven in [HUL93, Chap. 3.4]. However, the method was considered to be not implementable, since it requires a full subdifferential of the target function at each iterate generated by the method.

Concept of the Optimization Method LiPsMin

The idea of the optimization method LiPsMin is the minimization of composite piecewise differentiable objective functions via successive piecewise linearization. A Clarke stationary point of the arising local subproblems given by the piecewise linearization is determined by the algorithm PLMin. This algorithm is a crucial ingredient of LiPsMin. To assure that the piecewise linear model is bounded below, it is superimposed by a quadratic proximal term which is controlled by estimating the quality of the model.

The minimization of the local quadratic subproblem benefits from additional information obtained by exploiting the structure of the underlying piecewise linearization. The polyhedral structure of the argument space caused by the nondifferentiable points is used to analyze neighboring relations of these polyhedra. Hence, structure exploitation allows the identification of a descent trajectory along succeeding neighbor polyhedra. The availability of all components required by PLMin is mainly guaranteed by the abs-normal form which is an alternative representation of the piecewise linearization and allows an efficient evaluation of the very same linearization.

Content and Structure of this Thesis

This thesis is concerned with the development, implementation, and examination of an algorithm for the minimization of composite piecewise differentiable functions via successive piecewise linearization. To motivate the algorithm some nonsmooth sample problems are presented in Chap. 2.

The work is partitioned in three parts. Chap. 3 and Chap. 4 comprise an overview of basic aspects of nonsmooth analysis and optimization methods. In Chap. 3 the relevant concepts from convex and nonconvex nonsmooth analysis are summarized. A focus is set on Lipschitz continuous and piecewise differentiable functions. Furthermore, optimality conditions are discussed. In Chap. 4 several classes of gradient-based nonsmooth optimization methods are presented. The summery covers the fundamental subgradient and cutting-plane methods from the 1960s and 1970s as well as bundle and variable metric methods which were highly influenced by the previous methods. Furthermore, bundle and variable metric methods are the most common methods today and they are considered to be efficient and robust. Additionally, a gradient sampling method is introduced which represents another approach of nonsmooth optimization methods.

Chap. 5 and Chap. 6 are the centerpiece of this thesis. In these chapters the optimization method LiPsMin is presented. In Chap. 5 all components required by LiPsMin are introduced such as the generation of the piecewise linearization and its representation in abs-normal form as well as the evaluation of directionally active gradients. These components link the nonsmooth analysis as introduced in Chap. 3 with the goal of this thesis to develop a gray-box optimization method by allowing directional information instead of solely pointwise information as in classical black-box optimization schemes. The overall algorithm is finally presented in Chap. 6 where all components are pieced together. The algorithm consists of an inner and an outer loop whereby the outer loop successively generates the local models and controls the quadratic penalty term whereas the inner loop solves the sequence of local subproblems. Proving convergence of LiPsMin towards a Clarke stationary point tops the theoretical development of LiPsMin off.

In Chap. 7, the numerical performance of the new algorithm is investigated by comparing it with other state-of-the-art nonsmooth optimization software. Therefore,

a test set consisting of a combination of piecewise linear or piecewise smooth, and convex or nonconvex functions is defined. The majority of these test problems is scalable such that the performance of LiPsMin can be analyzed in terms of a growing number of optimization parameters and of occurring absolute value functions. Therefore, LiPsMin is compared with the bundle method MPBNGC and HANSO which combines a quasi-Newton method with a gradient sampling appraoch.

In Chap. 8 the results of this work are summarized and future research directions are discussed.

2

Sample Problems for Nonsmooth Optimization Problems

This preliminary chapter presents exemplary two established applications from the field of optimization theory that yield nonsmooth optimization problems of the form as the problems considered in this work.

2.1 Reformulation of Constrained Optimization Problems

A general formulation of a constrained optimization problem is given by

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ s.t. \quad &c_i(x) = 0, \quad i \in \mathcal{E}, \\ &c_i(x) \geq 0, \quad i \in \mathcal{I}, \end{aligned}$$

where the objective function f and the constraints c_i are all smooth, real-valued functions on a subset of $\mathbb{R}^n$, and $\mathcal{E}$ and $\mathcal{I}$ are two finite sets of indexes. This description of constrained optimization problems follows [NW06] where such problems are introduced in detail. First- and second-order optimality conditions of unconstrained problems are obtained by considering the Lagrangian function

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x)$$

where λ_i are the Lagrange multipliers. These multipliers can be considered as additional optimization parameters and they can be interpreted as a measure for the

sensitivity of the optimal objective function value to the presence of the corresponding constraint c_i.

A common strategy to solve constrained problems is the combination of the objective function and the constraints into a penalty function. Therewith, one obtains an unconstrained problem and can apply standard search techniques. A popular penalty function is the exact ℓ_1-penalty function of the form

$$\Phi(x;\mu) = f(x) + \mu \sum_{i\in\mathcal{E}} |c_i(x)| + \mu \sum_{i\in\mathcal{I}} \max\{0, -c_i(x)\}$$

where μ is the penalty parameter that punishes violations of the constraints. To solve the original constrained problem by the reformulated problem, one has to ensure that solutions (x^*, μ^*) of the unconstrained problem correspond with solutions (x^*, λ^*) of the original problem which usually holds for all sufficiently large μ. However, the occurring nonsmoothness may cause difficulties with regard to optimization methods which were originally designed for smooth unconstrained optimization.

2.2 Minimax Problems from Robust Optimization

Robust optimization problems result among others from decision making under uncertainties. There is a variety of decision models. Some discrete models were introduced in [KY97] including the following two models which induce nonsmooth optimization problems.

To define the models some notions are required that are introduced subsequently. Let S be the finite set of all potentially realizable input data scenarios over a prespecified planning horizon. Let D be the set of input data and D^s the instance of input data corresponding to scenario $s \in S$. Let X be the set of decision variables and F_s the set of all feasible decisions under a certain scenario s. The quality of the decision $X \in F_s$ is evaluated by applying the function $f(X, D^s)$.

Therewith, the two decision models can be defined. At first, the absolute robust decision X_A is defined as the one that minimizes the maximal total cost, among all

feasible decisions F_s over all realizable input data scenarios, i.e.,

$$\max_{s\in S} f(X_A, D^s) = \min_{X\in\bigcap_{s\in S} F_s} \max_{s\in S} f(X, D^s).$$

Another possible decision model is the robust deviation decision X_D that exhibits the best worst case deviation from optimality, among all feasible decisions F_s over all realizable input data scenarios, i.e.,

$$\max_{s\in S} \left(f(X_D, D^s) - f(X_s^*, D^s)\right) = \min_{X\in\bigcap_{s\in S} F_s} \max_{s\in S} \left(f(X, D^s) - f(X_s^*, D^s)\right).$$

The resulting minimax problems are typical for robust decision making and fit well into the framework of this thesis.

3

Nonsmooth Analysis

For the solution of nonsmooth optimization problems of the form

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a Lipschitz continuous but not necessarily differentiable function, it is important to understand nonsmooth analysis. It provides lots of theory about nonsmooth functions and their differentiability properties which allow us to define generalized derivative concepts and to derive optimality conditions.

In this chapter, an overview of important properties of convex, Lipschitz continuous and piecewise smooth functions and their subdifferentials will be given. Whereby Lipschitz continuous functions are going to be central, since they are the most general class of functions considered in this thesis. Based on them it will be explained how the behavior of piecewise smooth functions and their subdifferentials changes and simplifies, respectively. These properties of nonsmooth functions and their generalized derivatives, that are required in this work, will be introduced in the first three sections of this chapter. Subsequently, optimality conditions will be introduced. Using the subdifferentials of the previous sections, generalized criteria for stationary and minimal points will be given,

Throughout this chapter we will work in $\mathbb{R}^n$ with the Euclidean norm $\|\cdot\|_2$ and real-valued functions will be considered.

3.1 Convex Functions

At the beginning of the development of nonsmooth theory and optimization methods the class of convex functions was primarily considered. Therefore, many terms and concepts of the theory for Lipschitz continuous functions originated from the convex theory and thus it is reasonable that definitions and theorems concerning convex sets and functions will be recalled at the beginning of this chapter following [BKM14] and primarily [Roc70] by T.R. Rockafellar who summarized the field of convex analysis first and extended it.

Convex Sets and Cones

First, definitions relating to convex sets are summarized.

Definition 3.1 (Convex Set)**.** *Let U be a subset of $\mathbb{R}^n$. The set U is said to be convex, if $\lambda x + (1-\lambda)y \in U$ for all $x, y \in U$ and $\lambda \in [0,1]$.*

Convex sets bring lots of properties along, i.e., if $U_i \subseteq \mathbb{R}^n$ are convex sets for $i = 1, ..., m$, then their intersection $\cap_{i=1}^m U_i$ is again convex. Thus, one can write a polyhedral convex set as an intersection of finitely many closed half spaces of $\mathbb{R}^n$. Another useful concept is the convex combination which denotes the vector sum $\sum_{i=1}^k \lambda_i x_i$ with $x_i \in \mathbb{R}^n$, if $\lambda_i \geq 0$ for all $i = 1, ..., k$ and $\sum_{i=1}^k \lambda_i = 1$. The concept is used to define convex hulls.

Definition 3.2 (Convex Hull)**.** *The convex hull of a set $U \subseteq \mathbb{R}^n$ is*

$$conv\,(U) := \{x \in \mathbb{R}^n \mid x = \sum_{i=1}^{k} \lambda_i\, x_i,\ \sum_{i=1}^{k} \lambda_i = 1,\ x_i \in U,\ \lambda_i \geq 0,\ k > 0\}.$$

Typical sets considered in the convex theory are cones C that are defined as follows:

Definition 3.3 (Cone, Convex Cone)**.** *A set $C \subseteq \mathbb{R}^n$ is a cone if $\lambda x \in C$ for all $x \in C$ and $\lambda \geq 0$. Moreover, if C is convex, then it is called a convex cone.*

Definition 3.4 (Conic Hull)**.** *The conic hull of a set $U \subseteq \mathbb{R}^n$ is*

$$cone\,(U) := \{x \in \mathbb{R}^n \mid x = \sum_{i=1}^{k} \lambda_i\, x_i,\ x_i \in U,\ \lambda_i \geq 0,\ k > 0\}.$$

In [BKM14, Theorem 2.3], it is proven for both the convex hull and the conic hull that if U is a subset of $\mathbb{R}^n$, then

$$\text{conv}\,(U) \subseteq \bigcap_{\substack{U \subseteq \hat{U} \\ \hat{U}\ \text{convex}}} \hat{U} \quad \text{and} \quad \text{cone}\,(U) \subseteq \bigcap_{\substack{U \subseteq C \\ C\ \text{convex cone}}} C$$

Three well-known cones from convex analysis are the polar cone, the contingent cone and the normal cone that are defined below and are illustrated in Fig. 3.1.

Definition 3.5 (Polar, Contingent and Normal Cone)**.** *Let $U \subseteq \mathbb{R}^n$ be nonempty and convex.*

- *The polar cone of U is $U^\circ := \{y \in \mathbb{R}^n \mid y^\top x \leq 0$ for all $x \in U\}$.*
- *The contingent cone of U at $x \in U$ is given by*

$$K_U(x) := \{d \in \mathbb{R}^n \mid \text{There exist } t_i \downarrow 0 \text{ and } d_i \to d \text{ s.t. } x + t_i d_i \in U\}. \tag{3.1}$$

- *The normal cone of U at $x \in U$ is given by*

$$N_U(x) := K_U(x)^\circ = \{z \in \mathbb{R}^n \mid z^\top d \leq 0 \text{ for all } d \in K_U(x)\}. \tag{3.2}$$

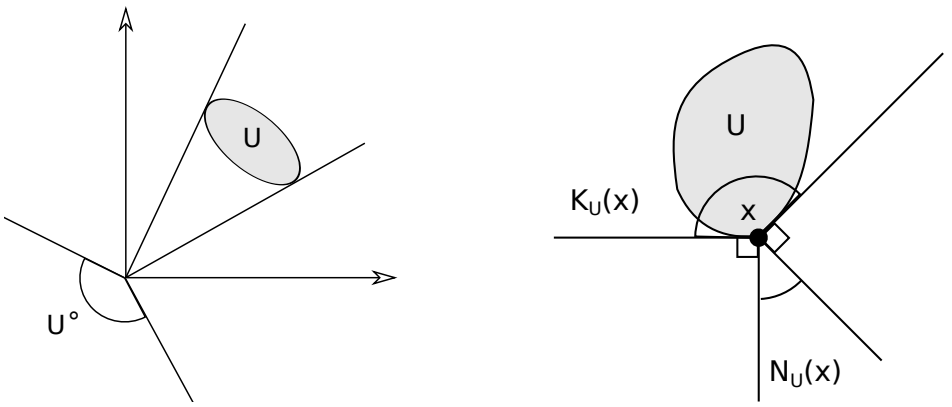
Convex Functions and Corresponding Subdifferentials

A function $f : \mathbb{R}^n \to \mathbb{R}$ is called convex, if all line segments between two points of the graph of f lie on or above that graph. The formal definition is given as follows:

Definition 3.6 (Convex Function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be convex if*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

whenever $x, y \in \mathbb{R}^n$ and $\lambda \in [0, 1]$.



(a) Polar cone of the set U

(b) Contingent and normal cone of the convex set U at x

Figure 3.1: Different cones corresponding to the set U

Due to the fact that convex functions are not necessarily differentiable, a generalized concept of gradients was developed and was first considered in detail in [Roc70].

Definition 3.7 (Subdifferential of Convex Function)**.** *The subdifferential of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ at $x \in \mathbb{R}^n$ is the set $\partial f(x)$ of vectors $\xi \in \mathbb{R}^n$ such that*

$$\partial f(x) = \left\{ \xi \in \mathbb{R}^n \mid f(y) \geq f(x) + \xi^\top (y - x) \quad \textit{for all} \quad y \in \mathbb{R}^n \right\}$$

The vectors $\xi \in \partial f(x)$ are called subgradients.

The epigraph of a function f is given by

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

and it was shown that epi f is convex if and only if f is convex, see [BKM14, Theorem 2.24]. Thus for each $\xi \in \partial f(x)$ the function $h(y) = f(x) + \xi^\top (y - x)$ is a supporting hyperplane to the convex set epi f at the point $(x, f(x))$.

Example 3.8. The subdifferential of the tangent function $f(x) = \tan |x|$ in $x = 0$ is given by $\partial f(x) = [-1, 1]$. In Fig. 3.2 the supporting hyperplanes h_1 for $\xi = 1$ and h_2 for $\xi = -1$ are illustrated by solid lines. The dashed lines signify all further hyperplanes given by the subdifferential.

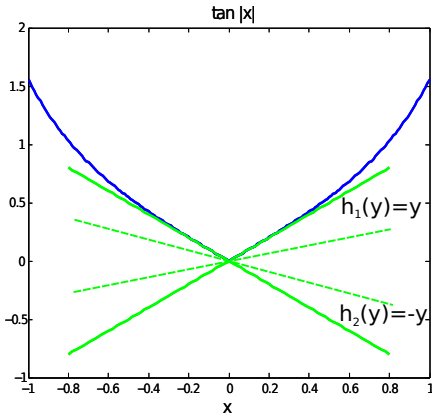


Figure 3.2: Supporting hyperplanes defined by the subdifferential $\partial f(x)$ in $x = 0$

In case the function f is convex and differentiable the function $h(y)$ represents the tangent of f in x, the subdifferential reduces to the gradient as stated below and thus the introduced subdifferential is a reasonable generalization of the classical derivative concept.

Proposition 3.9. *If $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable at $x \in \mathbb{R}^n$, then*

$$\partial f(x) = \{\nabla f(x)\}.$$

Proof. See [Roc70, Chap. 25], [BKM14, Theorem 2.29]. □

3.2 Lipschitz Continuous Functions

Since Lipschitz continuous functions which are not everywhere differentiable are considered in this work, a generalized concept of derivatives has to be used. In the fundamental book [Cla83] the convex theory introduced previously was extended to the more general class of Lipschitz continuous functions. Furthermore, it is shown that the generalized subdifferential can be written in terms of the limiting subdifferential. Both subdifferentials will be introduced in this subsection, as well as other necessary fundamental definitions and results, following [Cla83] and [BKM14].

Definition 3.10 (Lipschitz Continuity)**.** *Let $U \subseteq \mathbb{R}^n$ be an open subset, $f : U \to \mathbb{R}$ be a given function and $x \in U$. The function f is said to be locally Lipschitz*

continuous at x if there exists a scalar $L \geq 0$ and a positive number ϵ such that

$$\|f(y) - f(z)\| \leq L\|y - z\| \quad \forall z, y \in B_\epsilon(x) \cap U$$

holds where $B_\epsilon(x) := \{y \in \mathbb{R}^n \mid \|y - x\| < \epsilon\}$. If there exists a scalar $L \geq 0$ for all $z, y \in \mathbb{R}^n$ such that the inequality above holds, the function f is said to be Lipschitz continuous and L is called Lipschitz constant.

Note that one can prove that every convex function $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz continuous at any $x \in \mathbb{R}^n$, see, e.g., [BKM14, Chap. 2].

To define Clarke's subdifferential the generalized directional derivative as defined in [Cla83, Chap. 2] is required.

Definition 3.11 (Generalized Directional Derivative)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous at $x \in \mathbb{R}^n$ and $d \in \mathbb{R}^n$ a direction. The generalized directional gradient of f at x in the direction d, denoted $f^\circ(x; d)$, is defined as follows*

$$f^\circ(x; d) := \limsup_{\substack{y \to x \\ t \searrow 0}} \frac{f(y + td) - f(y)}{t}$$

where $y \in \mathbb{R}^n$ and $t \in \mathbb{R}$.

This definition does not require the existence of a limit. Additionally, it differs from the well known directional derivative

$$f'(x; d) := \lim_{t \searrow 0} \frac{f(x + td) - f(x)}{t}$$

in that the base point $y \in \mathbb{R}^n$ varies, which makes it interesting for objective functions that are not everywhere differentiable as will be explained later.

Let us recall some more basic definitions, that are useful when one considers generalized directional derivatives:

- A function $f : \mathbb{R}^n \setminus \{0\} \to \mathbb{R}$ is said to be *positively homogeneous of degree* $d \in \mathbb{R}$, if $f(tx) = t^d f(x)$ for $t > 0$.
- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *subadditive*, if $f(x + y) \leq f(x) + f(y)$ for all $x, y \in \mathbb{R}^n$.

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is *lower semi-continuous* at $x_0 \in \mathbb{R}^n$ if and only if $f(x_0) \leq \liminf_{x \to x_0} f(x)$.
- A function $f : \mathbb{R}^n \to \mathbb{R}$ is *upper semi-continuous* at $x_0 \in \mathbb{R}^n$ if and only if $f(x_0) \geq \limsup_{x \to x_0} f(x)$.

Remark: The function f is continuous, if f is lower and upper semi-continuous.

Proposition 3.12. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous with Lipschitz constant L at $x \in \mathbb{R}^n$. Then*

i) The function $d \mapsto f^\circ(x; d)$ is finite, positively homogeneous, and subadditive on $\mathbb{R}^n$, and satisfies $|f^\circ(x; d)| \leq L\|d\|$.

ii) The generalized directional derivative $f^\circ(x; d)$ is upper semi-continuous as a function of (x, d) and, as a function of d alone, is Lipschitz continuous with Lipschitz constant L on $\mathbb{R}^n$.

iii) $f^\circ(x; -d) = (-f)^\circ(x; d)$.

Proof. See [Cla83, Proposition 2.1.1]. □

One can now define Clarke's generalized gradient or Clarke's subdifferential, respectively. Its idea is that any positively homogeneous and subadditive functional on U majorizes some linear functionals on U by the Hahn-Banach Theorem where U is a Banach space. According to this and Prop. 3.12 there is at least one linear functional $\xi : U \to \mathbb{R}$ such that, for all $d \in U$, one has $f^\circ(x; d) \geq \langle \xi, d \rangle$. Thereby, ξ is bounded and belongs to the dual space U^* of continuous linear functionals on U. This leads to the following definition:

Definition 3.13 (Clarke's Subdifferential)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function at $x \in \mathbb{R}^n$. Clarke's subdifferential of f at x is a subset of $\mathbb{R}^n$ given by*

$$\partial_C f(x) := \{\xi \in \mathbb{R}^n : \ f^\circ(x; d) \ \geq \ \langle \xi, d \rangle \text{ for all } d \in \mathbb{R}^n\}.$$

Clarke's subdifferential is a multifunction which means that a point $x \in \mathbb{R}^n$ is assigned to a set $\partial_C f(x) \subseteq \mathbb{R}^n$. Furthermore, Clarke's subdifferential has some nice properties, as, for example, the properties in the following proposition.

Proposition 3.14. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous with Lipschitz constant L at a point $x \in \mathbb{R}^n$. Then*

i) $\partial_C f(x)$ is a nonempty, convex and compact subset of $\mathbb{R}^n$ and $\|\xi\| \leq L$ holds for every $\xi \in \partial_C f(x)$.

ii) For every $d \in \mathbb{R}^n$ one has $f^\circ(x; d) = \max\{\langle \xi, d\rangle : \xi \in \partial_C f(x)\}$.

Proof. See [Cla83, Proposition 2.1.2]. □

One of the most important properties of Clarke's generalized gradient is that due to Rademacher's theorem, see, e.g., [EG92], one can write Clarke's generalized gradient in terms of the limiting subdifferential $\partial_L f(x)$, that is defined in the following way:

Definition 3.15 (Limiting subdifferential)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous at $x \in \mathbb{R}^n$. The limiting subdifferential of f at x is the set*

$$\partial_L f(x) := \{\xi \in \mathbb{R}^n : \exists \{x_i\}_{i\in\mathbb{N}} \text{ with } x_i \notin \Omega_f \text{ s.t. } x_i \to x \text{ and } \nabla f(x_i) \to \xi\}$$

where Ω_f is the set of points where f is not differentiable.

Rademacher's theorem says that if $U \subseteq \mathbb{R}^n$ is an open subset and $f : U \to R$ is Lipschitz continuous, then f is differentiable almost everywhere in U. That is, the points in U at which f is not differentiable form a set Ω_f of Lebesgue measure zero. Clarke proved in [Cla83, Theorem 2.5.1] that one can write Clarke's subdifferential as the convex hull of the limiting subdifferential, i.e.,

$$\partial_C f(x) = \operatorname{conv}(\partial_L f(x)) \tag{3.3}$$

This insight is more than helpful for the algorithm introduced in this work, since it assures the computability of the subdifferential for piecewise smooth functions.

Derivative and Subderivative

As in the convex case, it can be shown that the subdifferential $\partial_C f(x)$ reduces to the derivative $\nabla f(x)$, if f is continuously differentiable at x. Thereby, Clarke's subdifferential is also a reasonable generalization in the nonconvex setting.

Proposition 3.16. *If $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable at $x \in \mathbb{R}^n$, then*

$$\partial_C f(x) = \{\nabla f(x)\}.$$

Proof. See [BKM14, Theorem 3.7]. □

Furthermore, it makes sense to consider strictly differentiable functions at this point, since this concept of differentiability is based on the generalized directional derivative which was used to define Clarke's subdifferential. Strict differentiability is defined as follows:

Definition 3.17 (Strict Differentiability)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a given function. Then f admits its strict derivative at x, a linear and bound operator denoted by $D_s f(x)$, if for each $d \in \mathbb{R}^n$*

$$\lim_{\substack{y \to x \\ t \searrow 0}} \frac{f(y + td) - f(y)}{t} = \langle D_s f(x), d \rangle \tag{3.4}$$

holds and provided that the convergence is uniform for d in compact sets.

Compared with the definition of standard differentiability, this definition also considers the point x as a limit of a sequence of points and therefore is more restrictive. As a consequence, every strictly differentiable function is differentiable, but the opposite direction does not hold, see Exam. 3.18.

Example 3.18. The continuous function

$$f : \mathbb{R} \to \mathbb{R}, \begin{cases} f(x) = x^2 \sin(\frac{1}{x}), & x \in \mathbb{R} \setminus \{0\} \\ f(x) = 0, & x = 0 \end{cases}$$

is differentiable at all $x \in \mathbb{R}$ but not strict differentiable. Considering the null sequences $\{y_n\}_{n \in \mathbb{N}}$ and $\{t_n\}_{n \in \mathbb{N}}$ defined as

$$y_n = \frac{1}{(n + \frac{3}{2})\pi} \quad \text{and} \quad t_n = \frac{1}{(n + \frac{1}{2})(n + \frac{3}{2})\pi}$$

and $d = 1$, the limit given in Eq. (3.4) does not converge towards a unique cluster point and thus, f is not strictly differentiable in $x = 0$.

The relations between the strict derivative and Clarke's generalized derivative is studied in the following proposition.

Proposition 3.19. *If $f : \mathbb{R}^n \to \mathbb{R}$ is strictly differentiable at x, then f is locally Lipschitz continuous at $x \in \mathbb{R}^n$ and $\partial_C f(x) = \{D_s f(x)\}$. Conversely, if f is Lipschitz continuous near x and $\partial_C f(x)$ reduces to a singleton $\{\xi\}$, then f is strictly differentiable at x and $D_s f(x) = \xi$.*

Proof. See [Cla83, Proposition 2.2.4]. □

Additionally, it is shown in [Cla83], that if f is locally Lipschitz continuous at $x \in \mathbb{R}^n$, then $\partial_C f(y)$ reduces to a singleton for every $y \in B_\epsilon(x)$ if f is continuously differentiable on $B_\epsilon(x)$. With Rademacher's theorem we can conclude that the subderivative of a Lipschitz function reduces to the standard derivative almost everywhere.

Geometric Interpretation

An alternative, geometric description of the generalized gradient can be gained from the study of cones. For the nonempty, convex set U, one considers the contingent cone of the set at $x \in U$ as given by Eq. (3.1). One can show that $K_U(x)$ is a closed convex cone, if U is a nonempty convex set. This property does not hold anymore, if U is a nonconvex set. Therefore, it is suggested in [Cla83, Chap. 2.4] to use the tangent cone

$$T_U(x) := \{y \in \mathbb{R}^n \mid d_U^\circ(x; y) = 0\}$$

for a nonconvex, nonempty set U at $x \in U$ in terms of the generalized directional derivative of the distance function which is given by

$$d_U(x) = \inf\{||x - y|| : \ y \in U\}.$$

Compared with the contingent cone the tangent cone is a closed convex set, if U is a nonempty set. If U is nonempty, one can show $T_U(x) \subseteq K_U(x)$, where equality holds whenever U is convex, see [BKM14, Theorem 3.27].

Since the above definition gives the impression that $T_U(x)$ depends on a particular norm, it is reasonable to find an equivalent expression that illustrates the norm independence of the tangent cone.

Theorem 3.20. *The tangent cone $T_U(x)$ of the nonempty set U at $x \in U$ can also be written as*

$$T_U(x) \equiv \{d \in \mathbb{R}^n \mid \forall t_i \to 0,\ t_i \in (0,\infty)\ \textit{and}\ \forall x_i \to x,\ x_i \in U, \\ \exists\, d_i \to d :\ x_i + t_i d_i \in U\}.$$

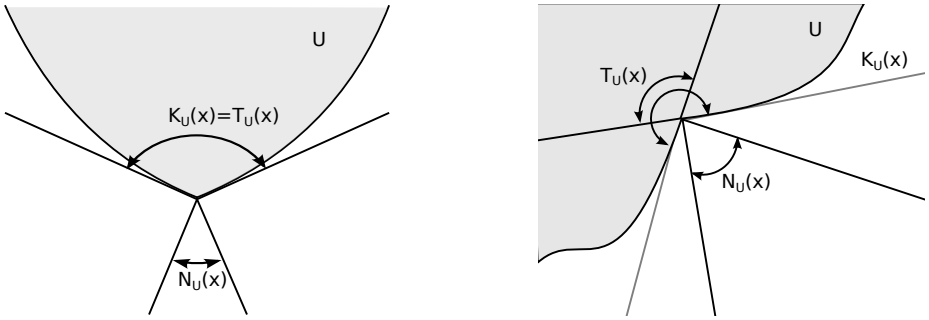
Proof. See [BKM14, Theorem 3.26]. □

Since the contingent cone $K_U(x)$ is not necessarily convex if U is nonconvex, the normal cone $N_U(x)$ for nonconvex sets U is defined via the tangent cone $T_U(x)$ as follows:

Definition 3.21 (Normal Cone of a Nonconvex Set)**.** *The normal cone of the nonempty set U at $x \in U$ is the set*

$$N_U(x) \equiv \{z \in \mathbb{R}^n \mid z^T d \leq 0\ \forall\ d \in T_U(x)\}.$$

In Fig. 3.3 the introduced cones are shown for both a convex and nonconvex set.



(a) Convex set U (b) Nonconvex set U

Figure 3.3: Contingent, tangent and normal cone

Finally, the following relationships between both the tangent cone of the epigraph

and the epigraph of the generalized directional derivative as well as Clarke's subdifferential and the normal cone can be shown.

Theorem 3.22. *If $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz continuous at x, then*

i) $T_{epi\ f}(x, f(x)) = epi\ f^\circ(x; \cdot)$.

ii) $\partial_C f(x) = \{\xi \in \mathbb{R}^n |\ (\xi, -1) \in N_{epi\ f}(x, f(x)\}$.

Proof. See [BKM14, Theorem 3.31, Theorem 3.32]. □

Rules of Calculus

For computing elements of Clarke's subdifferential, algorithmic differentiation (AD) will be used as will be explained in Chap. 5. One of the basic ideas of AD is to write a continuously differentiable function f as a composition of so called continuously differentiable elemental functions φ. The derivatives of these elemental functions are well known and the derivative of f is computed by the chain rule. Hence, it is important to have sharp rules of calculus as they are available for continuously differentiable functions also for the generalized case considered in this thesis. Unfortunately, most of these rules change in that they do not satisfy calculus rules sharply. Many of them can be found in [Cla83, Chap. 2.3] and [BKM14, Chap. 3.2.2]. In the following, some rules are given exemplarily.

- **Scalar multiples:** Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous at x. Then one has $\partial_C(\lambda f)(x) = \lambda \partial_C f(x)$ for all $\lambda \in \mathbb{R}$.
- **Finite sum:** Let $f_i : \mathbb{R}^n \to \mathbb{R}$ with $i = 1, ..., m$ be locally Lipschitz continuous at x. Then one has $\partial_C \left(\sum_{i=1}^m f_i\right)(x) \subseteq \sum_{i=1}^m \partial_C f_i(x)$.
- **Mean-Value Theorem:** Let $x, y \in \mathbb{R}^n$ with $x \neq y$, and suppose that f is locally Lipschitz continuous on an open set $U \subseteq \mathbb{R}^n$ such that the line segment $[x, y] \subseteq U$. Then there exists a point $u \in (x, y)$ such that

$$f(y) - f(x) \in \langle \partial_C f(u), y - x \rangle.$$

- **Chain rule:** Let f be such that $f = g \circ h$, where $h : \mathbb{R}^n \to \mathbb{R}^m$ is locally Lipschitz continuous at $x \in \mathbb{R}^n$ and $g : \mathbb{R}^m \to \mathbb{R}$ is locally Lipschitz continuous

at $h(x) \in \mathbb{R}^m$. Then f is locally Lipschitz continuous at $x \in \mathbb{R}^n$ and

$$\partial_C f(x) \subseteq \operatorname{conv}\{\partial_C h(x)^\top \partial_C g(h(x))\}.$$

- **Pointwise maxima:** Let $f_1, ..., f_m$ be locally Lipschitz continuous functions at x. Then the function $f : \mathbb{R}^n \to \mathbb{R}$ defined by

  $$f(x) := \max\{f_i(x) \mid i = 1, ..., m\}$$

  is locally Lipschitz continuous at x and

  $$\partial_C f(x) \subseteq \operatorname{conv}\{\partial_C f_i(x) \mid i \in \mathcal{I}(x)\}$$

  where $\mathcal{I}(x) := \{i \in \{1, ..., m\} \,|\, f_i(x) = f(x)\}$.
- **Product rule:** Let f_1 and f_2 be locally Lipschitz continuous at $x \in \mathbb{R}^n$. Then $f_1 f_2$ is locally Lipschitz at $x \in \mathbb{R}^n$, and one has

  $$\partial_C (f_1 f_2)(x) \subseteq f_2(x)\partial_C f_1(x) + f_1(x)\partial_C f_2(x).$$

Sharp calculus rules lose validity also for further rules, i.e., the evaluation of partial generalized gradients. Because of that it is necessary to find how to guarantee that only elements of the considered subdifferential can be computed. Sharp calculus rules are guaranteed for lexicographic functions as introduced in [Nes05]. It was shown in [KB15] that piecewise differentiable functions in the sense of [Sch12] are lexicographic functions. Within the scope of this thesis, directionally active gradients as defined in Chap. 5.4 are used to compute guaranteed elements of Clarke's subdifferential.

3.3 Piecewise Differentiable Functions

The objective functions of the minimization problem considered in this thesis were defined to be piecewise differentiable functions. S. Scholtes addressed this class of functions in depth in his habilitation thesis in 1994 that was reprinted in 2012, see [Sch12]. Parts of this work will be summarized in the following section. Especially characteristics of piecewise affine and piecewise smooth functions will be introduced.

The optimization algorithm developed in Chap. 6 uses piecewise affine functions to build a local model of the piecewise smooth objective function and thus, they are an important component of the introduced algorithm. Subsequently, piecewise smooth functions in general will be discussed.

3.3.1 Piecewise Affine Functions

Piecewise affine functions possess some very useful properties that will be examined subsequently as the polyhedral decomposition of the domain, the representation by superposition of finitely many minimum and maximum functions and finally a beneficial description of the subdifferential. Since the polyhedral structure plays a decisive role when considering piecewise affine functions, this section starts with an overview of important definitions of polyhedral theory.

Polyhedral Sets and Polyhedral Cones

In the following, important definition and results from [Sch12] that are required to describe a polyhedron will be summarized. Some terms needed for the discussion of polyhedral theory were already introduced in previous sections, as the convex hull, the convex cone and the normal cone. The affine hull of U given by

$$\text{aff}(U) = \{\sum_{i=1}^{m} \lambda_i x_i \mid m \in \mathbb{N},\ x_i \in U,\ \lambda_i \in \mathbb{R},\ \sum_{i=1}^{m} \lambda_i = 1\}$$

is required to define the relative interior of U.

Definition 3.23 (Relative Interior)**.** *A point $x \in U$ is called a relative interior point of U if there exists a number $\epsilon > 0$ such that every point $y \in \text{aff}(U)$ with $\|y - x\| < \epsilon$ is contained in U. The set of all relative interior points of U, denoted relint(U), is called relative interior of U.*

Thus, central definitions of this subsection are the following terms:

Definition 3.24 (Polyhedron, Polytope)**.** *A nonempty set $P \subseteq \mathbb{R}^n$ is called a polyhedron if there exists a real $m \times n$-matrix A and a $b \in \mathbb{R}^m$ such that*

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}.$$

A compact polyhedron is called a polytope.

Definition 3.25 (Polyhedral Cone)**.** *A nonempty set $C \subseteq \mathbb{R}^n$ is called a polyhedral cone if there exists a $m \times n$-matrix A such that*

$$C = \{x \in \mathbb{R}^n \mid Ax \leq 0\}.$$

The normal cone of the polyhedral cone C at the origin is characterized by the Farkas' lemma as quoted in [Sch12, Lemma 2.1.1] by

$$N_C(0) = \text{cone}\{a_i \mid i \in \{1, ..., m\}\}$$

where $a_i \in \mathbb{R}^n$, $i = 1, ..., n$, are the rows of A. Furthermore, via the Farkas-Minkowski-Weyl theorem one can describe the normal cone of a polyhedron $P = \{y \in \mathbb{R}^n \mid a_i^\top y \leq b_i,\ i = 1, ..., m\}$ with $x \in P$ by

$$N_P(x) = \text{cone}\{a_i \mid i \in \{1, ..., m\},\ a_i^\top x = b_i\}.$$

Considering the faces of a polyhedron allows us to describe the polyhedron in more detail. Therefore, the following definitions are introduced.

Definition 3.26 (Max-Face)**.** *Let $U \in \mathbb{R}^n$ be a closed convex set. The mapping F_U assigns to each linear functional in $\mathbb{R}^n$ the set of all maximizers $x \in U$ of the linear functional over U, i.e.,*

$$F_U(y) = \{x \in U \mid y^\top x \geq y^\top z \text{ for every } z \in U\}.$$

The set $F_U(y)$ is called max-face of the set U corresponding to the vector $y \in \mathbb{R}^n$.

Definition 3.27 (Face Lattice, Face, Collection of Index Sets)**.** *Let $P \subseteq \mathbb{R}^n$ be a polyhedron of the form $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.*

- *The collection of all max-faces of P together with the empty set is called the face lattice of P.*
- *The elements of the face lattice are called faces. If a face is nonempty and does not coincide with P, it is called a proper face.*
- *The collection of index sets $\mathcal{I}(A, b)$ is defined by*

$$\mathcal{I}(A,b) = \{I \subseteq \{1,...,m\} \mid \text{There exists a } x \in \mathbb{R}^n \text{ with } a_i^\top x = b_i,\ i \in I,\ a_j^\top x < b_j,\ j \in \{1,...,m\} \setminus I\}$$

and for every index set $I \subseteq \{1,...,m\}$ a corresponding polyhedron is given by

$$P_I = \{x \in \mathbb{R}^n \mid a_i^\top x = b_i,\ i \in I,\ a_j^\top x \leq b_j,\ j \in \{1,...,m\} \setminus I\}.$$

Hence, faces of P can be represented by index sets and thereby, further properties of the elements of the face lattice can be shown.

Proposition 3.28.

- *A subset $\tilde{P} \subseteq P$ is a max-face of P if and only if there exists an index set $I \in \mathcal{I}(A, b)$ such that $\tilde{P} = P_I$.*
- *Any two faces P_I and P_J corresponding to distinct index sets $I,\ J \in \mathcal{I}(A, b)$ are distinct.*
- *$I \cap J \in \mathcal{I}(A, b)$ for any two index sets $I,\ J \in \mathcal{I}(A, b)$.*

Proof. See [Sch12, Proposition 2.1.3]. □

The face lattice and the collection of index sets will be used in Subsection 5.4.2 to define a partially ordering via the faces of a polyhedron.

Properties and Representations of Piecewise Affine Functions

Subsequently, piecewise affine functions will be defined. Furthermore, two alternative representations will be discussed and an important result from [Sch12] regarding the decomposition of the domain caused by nondifferentiable points of piecewise

affine functions will be introduced.

Definition 3.29 (Piecewise Affine Function, Selection Function)**.** *A continuous function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called piecewise affine if there exists a finite set of affine functions $f_i(x) = A_i x + b_i$, $i = 1, ..., k$, such that the inclusion $f(x) \in \{f_1(x), ..., f_k(x)\}$ holds for every $x \in \mathbb{R}^n$.*

The affine functions $f_i(x) = A_i x + b_i$, $i = 1, ..., k$ are called selection functions.

The function f is called piecewise linear, if there exist a corresponding set of linear selection functions.

It can be shown that piecewise affine functions are Lipschitz continuous.

Proposition 3.30. *Every piecewise affine function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous. If (A_1, b_1),...,(A_k, b_k) is a collection of matrix-vector pairs corresponding to f, then $\max\{|||A_1|||, ..., |||A_k|||\}$ is a Lipschitz constant of f where $|||A|||$ is the operator norm defined as*

$$|||A||| = \max_{x \neq y} \frac{\|Ax - Ay\|}{\|x - y\|}.$$

Proof. See [Sch12, Proposition 2.2.7]. □

Subsequently we will assume that $m = 1$ and thus, we consider the affine selection functions $f_i(x) = a_i^\top x + b_i$ for $i = 1, ..., k$.

There are several generalized ways to represent piecewise affine functions. These representations can be used to illustrate several properties of piecewise affine functions and in some cases also to exploit these properties.

A notable representation of piecewise affine functions is the max-min representation, since it is not obvious that every real-valued, piecewise affine function can be expressed as a superposition of finitely many minimum and maximum functions. Moreover, this means that every piecewise affine function can be written in terms of the absolute value function.

Proposition 3.31. *If $f : \mathbb{R}^n \to \mathbb{R}$ is piecewise affine with affine selection functions $f_1(x) = a_1^\top x + b_1, ..., f_k(x) = a_k^\top x + b_k$, then there exist a finite number of index sets*

$M_1, ..., M_l \subseteq \{1, ..., k\}$ *such that*

$$f(x) = \max_{1 \leq i \leq l} \min_{j \in M_i} a_i^\top x + b_i.$$

Proof. See [Sch12, Proposition 2.2.2]. □

The drawback of this representation is that it may be very difficult to transform arbitrary piecewise affine functions into the max-min representation.

Another possibility described in [Sch12] is to write a piecewise affine function in terms of its set of selection functions $a_i^\top x + b_i$ for $i = 1, ..., k$. Let us assume that these selection functions are mutually distinct and consider the sets $p_i = \{x \in \mathbb{R}^n \mid f(x) = a_i^\top x + b_i\}$ for $i = 1, ..., k$. Since f is continuous, the sets p_i are closed and since f is piecewise affine and the selection functions are mutually distinct the union of all sets p_i covers $\mathbb{R}^n$ and the collection of all sets p_i with nonempty interior is a decomposition of $\mathbb{R}^n$. In [Sch12, Proposition 2.2.3] it is proven that every piecewise affine (piecewise linear) function $f : \mathbb{R}^n \to \mathbb{R}$ admits a corresponding polyhedral (conical) decomposition of $\mathbb{R}^n$. How to describe this structure for the purpose of this thesis and how it can be efficiently exploited for the introduced optimization algorithm will be explained in Subsection 5.4.2.

3.3.2 Piecewise Smooth Functions

Finally, the objective functions targeted by the considered optimization problem will be introduced. As before we follow [Sch12] for this brief introduction and start with the definition of piecewise smooth functions.

Definition 3.32 (Continuous Selection, Piecewise Differentiable Function)**.** *Let $U \subseteq \mathbb{R}^n$ and $f_i : U \to \mathbb{R}^m$, $i = 1, ..., k$, be a collection of continuous functions. A function $f : U \to \mathbb{R}^m$ is said to be a continuous selection of functions $f_1, ..., f_k$ on the set $O \subseteq U$ if f is continuous on O and $f(x) \in \{f_1(x), ..., f_k(x)\}$ for every $x \in O$.*

A function $f : U \to \mathbb{R}^m$ defined on an open set $U \subseteq \mathbb{R}^n$ is called piecewise differentiable, if for every $x_0 \in U$ there exists an open neighborhood $O \subseteq U$ and a finite number of continuously differentiable functions $f_i : O \to \mathbb{R}^m$, $i = 1, ..., k$, such that f is a continuous selection of $f_1, ..., f_k$ on O.

Subsequently, we will again only consider the case $m = 1$.

Since only real-valued functions are considered throughout this thesis, one can show that the result of a variety of operations on piecewise differentiable functions is again a piecewise differentiable function, as for scalar multiplication, finite sum, and pointwise maximum or minimum. This also holds for the superposition $f \circ g$ of two piecewise differentiable functions.

To describe piecewise differentiable functions in more detail, the active index set as well as the essentially active index set are useful.

Definition 3.33 (Active Index Set, Essentially Active Index Set)**.** *Given a set of selection functions $f_1, ..., f_k$ for a piecewise differentiable function f at a point x_0, the active set at the point x_0 is given by*

$$I_f(x_0) = \{i \in \{1, ..., k\} \mid f(x_0) = f_i(x_0)\}.$$

The selection functions f_i, $i \in I_f(x_0)$, are called active selection functions at x_0.

Furthermore, the set of essentially active indices is given by

$$I_f^e(x_0) = \{i \in \{1, ..., k\} \mid x_0 \in cl(int\{x \in U \mid f(x) = f_i(x)\})\}.$$

A selection function f_i is called essentially active at x_0 if $i \in I_f^e(x_0)$.

For the algorithm developed in this thesis both index sets play an important part but since the identification of the active functions will be realized in an adapted manner, we will not go into detail at this point any further.

Lipschitz Continuity and Subdifferentials of Piecewise Smooth Functions

Concluding this section about piecewise differentiable functions, two important results proven in [Sch12] will be presented. First, it is shown in that book that piecewise smooth functions as introduced in this section are Lipschitz continuous.

Proposition 3.34. *Every piecewise differentiable function is locally Lipschitz continuous. A Lipschitz constant in a neighborhood of x_0 is given by the maximum of the Lipschitz constants of the selection functions.*

Proof. See [Sch12, Corollary 4.1.1]. □

This obviously means that the theory of Lipschitz continuous functions that was introduced in subsection 3.2 can be applied.

The second result is a description of the Clarke's subdifferential via the limiting subdifferential by integrating the additional properties of piecewise smooth functions.

Proposition 3.35. *If $U \subseteq \mathbb{R}^n$ open and $f : U \to \mathbb{R}$ is a piecewise differentiable function with C^1 selection functions $f_i : O \to \mathbb{R}$, $i = 1, ..., k$ at $x_0 \in O \subseteq U$, then*

$$\partial_C f(x_0) = conv(\, \nabla f_i(x_0) \ : \ i \in I_f^e(x_0)\,).$$

Proof. See [Sch12, Proposition 4.3.1]. □

Thus, the limiting gradient of a piecewise smooth function consists of finitely many gradients each corresponding to an essentially active selection function. This is a remarkable property of the limiting subdifferential, since it ensures that it can be computed entirely. Additionally, Clarke's subdifferential is available as the convex hull of the limiting subdifferential.

3.4 Optimality Conditions

In this concluding section, the topics of nonsmooth analysis and optimization will be linked by defining local and global minima and by presenting generalized optimality conditions for nonsmooth optimization problems. The considered unconstrained optimization problem is of the form

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

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous and piecewise smooth. In general, minimal points and Clarke stationary points are defined as follows:

- A point x^* is a global minimizer of f if $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.

- A point x^* is a local minimizer of f if there is a neighborhood U of x^* such that $f(x^*) \leq f(x)$ for all $x \in U$.
- A point x^* is a strict local minimizer of f if there is a neighborhood U of x^* such that $f(x^*) < f(x)$ for all $x \in U$ with $x \neq x^*$.
- A point $x \in \mathbb{R}^n$ is a Clarke stationary point of f if it satisfies $0 \in \partial_C f(x)$.

First order necessary conditions for the optimization problem (P) are given in the following proposition.

Proposition 3.36. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function at $x^* \in \mathbb{R}^n$. If f attains a local minimum at x^*, then*

$$0 \in \partial_C f(x^*) \quad \text{and} \quad f^\circ(x^*; d) \geq 0 \quad \text{for all} \quad d \in \mathbb{R}^n.$$

Proof. See [BKM14, Theorem 4.1]. □

If the objective function is also convex and $x^* \in \mathbb{R}^n$ is a Clarke stationary point, one obtains that x^* is a global minimizer of f. This can easily be shown by applying the subgradient inequality from Def. 3.7. Furthermore, sufficient optimality conditions can be formulated, if f is convex.

Proposition 3.37. *If the function $f : \mathbb{R}^n \to \mathbb{R}$ is convex, then f attains its global minimum at $x^* \in \mathbb{R}^n$ if and only if at least one of the following conditions holds:*

i) $0 \in \partial_C f(x^*)$,

ii) $f'(x^*; d) \geq 0$ *for all* $d \in \mathbb{R}^n$.

Proof. See [BKM14, Theorem 4.2]. □

4

Nonsmooth Optimization Methods

Since the 1960s various approaches for solving nonsmooth optimization problems were developed. The purpose of this chapter is to give a brief overview of methods that pointed the way ahead as subgradient and cutting-plane methods and state-of-the-art methods as bundle methods, variable metric methods and gradient sampling methods. This overview does not claim to be complete. Nevertheless, it explains various important ideas from nonsmooth optimization considering representative popular methods.

Subgradient methods generalizing descent methods by replacing the gradient by an arbitrary subgradient were introduced amongst others by N.Z. Shor and will be summarized in Sec. 4.1. The standard cutting-plane method was presented by J.E. Kelley and will be explained in Sec. 4.2. Cutting-plane methods are the basis for bundle methods which are wide-spread today and are outlined in Sec. 4.3. They are supposed to be very efficient and robust. Since there exists a huge variety of bundle methods with different priorities and characteristics, the brief introduction will be focused on a proximal bundle method proposed by M.M. Mäkelä and P. Neittaanmäki with similar requirements as the optimization method developed in this thesis.

The idea of subgradient methods was also refined in such a way that methods originally developed for smooth optimization problems were generalized for the nonsmooth case. Promising methods of smooth optimization were adapted such as variable metric methods and trust region methods. In Sec. 4.4, a variable metric method will be presented which utilizes the BFGS method that was developed by A.S. Lewis and Micheal L. Overton and which again assumes similar properties of the objective functions as the method developed in this thesis.

In the 2000s a new approach was presented by J.V. Burke, A.S. Lewis and M.L. Overton in [BLO05]. They introduced gradient sampling methods which do not require any subgradient information and will be summarized in Sec. 4.5.

4.1 Subgradient Methods

Subgradient methods were developed since the 1960s. An early overview of subgradient methods can be found in [Sho79, Chap. 2], whereas this brief introduction follows [BKM14, Chap. 10]. The idea of subgradient methods is to generalize gradient descent methods from smooth optimization by replacing the gradient by an arbitrary subgradient. Thus, one obtains the iteration formula

$$x_{k+1} = x_k - t_k \frac{\xi_k}{\|\xi_k\|} \tag{4.1}$$

where $\xi_k \in \partial_C f(x_k)$ is a subgradient of f at the iterate x_k and $t_k > 0$ is a step multiplier. A difficulty is the definition of a termination criterion, since the sequence of subgradient ξ_k does not necessarily converge to 0. Another disadvantage is that contrary to the gradient descent method one can not guarantee that the search direction obtained from the subgradient causes descent. As a consequence, standard line searches can not be applied. To achieve some convergence statements further assumptions concerning the step size have to be made. In [Sho79, Chap. 2.2] it was proven that the subgradient method convergences globally for a convex function and step multipliers satisfying

$$\lim_{k\to\infty} t_k = 0 \quad \text{and} \quad \sum_{k=1}^{\infty} t_k = +\infty.$$

Under certain additional assumptions a linear rate of convergence can be proven. To improve the rate of convergence several other algorithms were developed such as Shor's r-algorithm with space dilation described in [Sho79, Chap. 3]. Its idea is to combine two sequential subgradients and thereby, to obtain additional information.

4.2 Cutting-Plane Methods

The standard cutting plane method was introduced by J.E. Kelley in [Kel60]. As in the previous section this brief introduction follows [BKM14, Chap. 11]. It considers optimization problems of the form

$$\min_{x \in X} f(x)$$

where $X \subseteq \mathbb{R}^n$ is a nonempty, closed convex set and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function. The idea of the method is to approximate the objective function from below by a piecewise affine function

$$\hat{f}^k(x) = \max_{j=0,\dots,k} \{f(x_j) + \xi_j^\top (x - x_j)\} \tag{4.2}$$

where x_k is the current iterate and the other x_j with $j = 0, ..., k-1$ are auxiliary points. One assumes that for each point x_j, $j = 0, ..., k$, a subgradient $\xi_j \in \partial_C f(x_j)$ is available. Thus, for each $j = 0, ..., k$ and for all $x \in X$ the inequality $f(x) \geq f(x_j) + \xi_j^\top (x - x_j)$ holds and one obtains that

$$f(x) \geq \max_{j=0,\dots,k} \{f(x_j) + \xi_j^\top (x - x_j)\} \quad \text{for all} \quad x \in X.$$

To identify a new iterate, one considers the minimization problem

$$\min_{x \in X} \max_{j=0,\dots,k} \{f(x_j) + \xi_j^\top (x - x_j)\} - f(x_k)$$

where f is replaced by its approximation $\hat{f}^k$. Subtracting the function value $f(x_k)$ from the objective function allows the reformulation of this nondifferentiable optimization problem into a linearly constrained problem of the form

$$\min_{\substack{v \in \mathbb{R} \\ x_k + d \in X}} v \qquad \text{s.t.} \quad -\alpha_j + \xi_j^\top d \leq v \quad \forall j = 0, ..., k \tag{4.3}$$

where $d = x - x_k$ and $\alpha_j := f(x_k) - f(x_j) - \xi_j^\top (x_k - x_j)$ is the linearization error. Thus, the new iterate $x_{k+1} = x_k + d$ is given by the solution of (4.3). By adding the corresponding cutting plane to the approximation (4.2), one obtains a more precise

model in such a way that $\hat{f}^k(x) \leq \hat{f}^{k+1}(x)$. A possible termination condition is given by the linearization error $\alpha_k \leq \epsilon$.

The initial iteration of the cutting-plane method can cause difficulties. If the set X is not chosen properly, the approximation $\hat{f}^0$ might not be bounded below. Thus, the choice of an appropriate initial set X is crucial for the success of the method. Assuming that the set X is chosen properly, the graph of the model $\hat{f}^k$ approximates the original objective function f more accurately from below with each iteration and thereby, the global convergence of the method can be guaranteed as shown in [Kel60]. However, the convergence results are rather poor in practice.

To remedy these disadvantages of the cutting-plane method first bundle methods were developed as for instance by Kiwiel, see [Kiw85].

4.3 Bundle Methods

Nowadays, some of the most common methods in nonsmooth optimization are bundle-type methods. The idea of these methods is to exploit the previous iterations by gathering the corresponding subgradient information in a bundle. This is an important difference to subgradient-type methods that only use local information at the current iterate. By looking at a bundle of subgradients it is possible to define and implement stopping criteria, which is an additional advantage compared with subgradient-type methods. Bundle methods are quite similar to cutting plane methods. However, in contrast to cutting plane methods, bundle methods do not only gather subgradient information but also remove subgradients from the bundle due to certain heuristics. First bundle-type methods for convex, locally Lipschitz functions were developed in the 1970s and 1980s. These approaches were refined over the years and extended to nonconvex and constraint optimization problems.

The proximal bundle method for nonconvex constrained optimization introduced by M.M. Mäkelä and P. Neittaanmäki in [MN92, Chap. 3] is a well-known representative of these extensions. Thus, the ideas of bundle methods will be described by reference

of this example in the following. They consider the problem

$$\left.\begin{array}{lll} \min f(x) \quad s.t. & F_i(x) \leq 0, & \text{for} \quad i = 1, ..., m_F, \\ & Cx \leq b, \quad C \in \mathbb{R}^{m_C \times n}, & b \in \mathbb{R}^{m_C}, \\ & x_{\min} \leq x \leq x_{\max}, & x_{\min},\ x_{\max} \in \mathbb{R}^n \end{array}\right\} \tag{4.4}$$

where f and F_i for $i = 1, ..., m_F$ are locally Lipschitz functions defined on $\mathbb{R}^n$. The rows of C are denoted by C_i. The total constraint function is defined by

$$F(x) = \max\{F_i(x) \mid i = 1, ..., m_F\}.$$

Suppose that problem (4.4) satisfies the Cottle constraint qualification as defined below, and that the feasible set $G = G_F \cap G_C = \{x \mid F(x) \leq 0\} \cup \{x \mid Cx \leq b\}$ is nonempty.

Definition 4.1 (Cottle constraint qualification)**.** *The problem (4.4) is said to satisfy the Cottle constraint qualification at x if either $F(x) < 0$ or there do not exist any $\nu_i \geq 0$ for $i \in I := \{1, ..., m_C\}$ such that $\nu_i(C_i^T x - b_i) = 0$ and*

$$0 \in \partial F(x) + \sum_{i \in I} \nu_i C_i.$$

Assume that the subgradients $\xi^f \in \partial f(x)$, $\xi^F \in \partial F(x)$ and function values $f(x)$, $F(x)$ can be evaluated. One defines an improvement function

$$H(x; y) := \max\{f(x) - f(y), F(x)\} \qquad \text{for all } x \in \mathbb{R}^n$$

at $y \in \mathbb{R}^n$. If the current iterate $x_k \in \mathbb{R}^n$ is nonoptimal, we would like to find the descent direction $d_k \in \mathbb{R}^n$ that solves the linearly constraint problem

$$\min H(x_k + d; x_k) \qquad \text{s.t.} \qquad x_k + d \in G_C. \tag{4.5}$$

For the moment, we will suppose that problem (4.4) is convex. One assumes that we have at the current iterate x_k the auxiliary points $y_j \in \mathbb{R}^n$ and the subgradients $\xi_j^f \in \partial f(y_j)$ for $j \in J_f^k \subset \{1, ..., k\}$ and $\xi_j^F \in \partial F(y_j)$ for $j \in J_F^k \subset \{1, ..., k\}$ where the index sets J_f^k and J_F^k are assumed to be nonempty. With these subgradients one

defines the linearizations at $x \in \mathbb{R}^n$ by

$$\begin{aligned} \bar{f}_j(x) &:= \bar{f}(x; y_j) = f(y_j) + (\xi_j^f)^T(x - xj) \qquad \text{for all} \quad j \in J_f^k \\ \bar{F}_j(x) &:= \bar{F}(x; y_j) = F(y_j) + (\xi_j^F)^T(x - xj) \qquad \text{for all} \quad j \in J_F^k \end{aligned}$$

and the polyhedral approximation of $H(\cdot, x_k)$ by

$$\hat{H}^k(x) := \max\{\hat{f}_k(x) - f(x),\ \hat{F}^k(x)\}$$

with $\hat{f}^k(x) := \max\{\bar{f}_j(x) \mid j \in J_f^k\}$ and $\hat{F}^k(x) := \max\{\bar{F}_j(x) \mid j \in J_F^k\}$.

Analogous to the cutting plane method one replaces the original improvement function by its approximation $\hat{H}_k$. Therewith, one obtains the following approximation of problem (4.5)

$$\min \hat{H}(x_k + d) + \frac{u_k}{2}\|d\|^2 \qquad \text{s.t.} \qquad C(x_k + d) \leq b, \tag{4.6}$$

where $u_k > 0$ is a weighting parameter and the quadratic penalty term ensures that the problem is bounded below. Hence, a solution of the problem exists. The weighting parameter is updated by the safeguarded quadratic interpolation technique that was introduced by Kiwiel in [Kiw90]. This problem can be rewritten as a (differentiable) quadratic problem of the form

$$\left.\begin{aligned} \min v + \tfrac{u_k}{2}\|d\|^2 \qquad s.t. \quad & -\alpha_{f,j}^k + (\xi_j^f)^T d \leq v, \quad \text{for all} \quad j \in J_f^k, \\ & -\alpha_{F,j}^k + (\xi_j^F)^T d \leq v, \quad \text{for all} \quad j \in J_F^k, \\ & -\alpha_{C,i}^k + C_i^T d \leq 0, \qquad \text{for all} \quad i \in I. \end{aligned}\right\} \tag{4.7}$$

by defining the so-called linearization errors $\alpha_{f,j}^k := f(x_k) - \bar{f}_j(x_k)$ for $j \in J_f^k$, $\alpha_{F,j}^k := -\bar{F}_j(x_k)$ for $j \in J_F^k$ and $\alpha_{C,i}^k := -C_i^T x_k + b_i$ for $i \in I$.

If one always added the index of each iteration, problems with storage would be inevitable after a huge number of iteration. The authors use the subgradient aggregation strategy by [Kiw85] to choose the index sets J_f^k and J_F^k. This strategy allows the user to fix a maximal number of indexes M_ξ stored in J_f^k and J_F^k.

An open question is what difficulties occur if the problem (4.4) is nonconvex. It turns out that the linearization error is no sufficient measure anymore, since the

approximation of the improvement function $H(\cdot, x_k)$ is no lower approximation of the target function anymore and thus, the linearization errors are not necessarily greater or equal to zero. Because of this, one replaces the linearization error in problem (4.7) by the subgradient locality measure

$$\begin{aligned} \beta_{f,j}^k &:= \max\{|\alpha_{f,j}^k|,\ \gamma_f \cdot (s_j^k)^2\}, && \text{for all} \quad j \in J_f^k \\ \beta_{F,j}^k &:= \max\{|\alpha_{F,j}^k|,\ \gamma_F \cdot (s_j^k)^2\}, && \text{for all} \quad j \in J_F^k \end{aligned}$$

where $\gamma_f > 0$ and $\gamma_F > 0$ are user-defined distance measure parameters that weight the distance measure

$$s_j^k := \begin{cases} \|x_j - y_j\| + \sum_{i=j}^{k-1} \|x_{i+1} - x_i\| & \text{for} \quad j = 1, ..., k-1 \\ \|x_k - y_k\| & \text{for} \quad j = k. \end{cases}$$

Afterwards, one would like to compute a step size $t_k \in (0, 1]$ such that the step multiplier approximately minimizes the objective function f along a given direction d and that the resulting iterate is an element of the feasible set G. In [MN92, Chap. 3.2] a two–point line search is introduced. It assumes that $m_L \in (0, \frac{1}{2})$, $m_R \in (m_L, 1)$ and $\bar{t} \in (0, 1]$ are fixed parameters. The first step of this line search strategy is to find the largest number $t_L^k \in [0, 1]$ such that

$$\begin{aligned} &\text{a)} \quad f(x_k + t_L^k d_k) \le f(x_k) + m_L t_L^k v_k, && \text{b)} \quad F(x_k + t_L^k d_k) \le 0, \\ &\text{c)} \quad C(x_k + t_L^k d_k) \le b, && \text{d)} \quad t_L^k \ge \bar{t}. \end{aligned}$$

If such a parameter exists one takes a long serious step

$$x_{k+1} := x_k + t_L^k d_k \qquad \text{and} \qquad y_{k+1} := x_{k+1}.$$

In this case one achieves a significant descent and one sets $\xi_{k+1}^f \in \partial f(x_{k+1})$. If requirements a) - c) hold but $0 < t_L^k < \bar{t}$ then we take a short serious step

$$x_{k+1} := x_k + t_L^k d_k \qquad \text{and} \qquad y_{k+1} := x_k + t_R^k d_k$$

and if $t_L^k = 0$ we take a null step

$$x_{k+1} := x_k \qquad \text{and} \qquad y_{k+1} := x_k + t_R^k d_k$$

where $t_R^k > t_L^k$ is such that $-\beta_{f,k+1}^{k+1} + (\xi_{k+1}^f)^T d_k \geq m_R v_k$. In these two cases there are discontinuities in the gradient of f. The additional requirement $-\beta_{f,k+1}^{k+1} + (\xi_{k+1}^f)^T d_k \geq m_R v_k$ in the null step ensures that x_{k+1} and y_{k+1} lie on opposite sites of a discontinuity of the gradient and thus the new subgradient $\xi_{k+1}^f \in \partial f(y_{k+1})$ will force a significant modification of the next search direction finding problem.

The last major component is the stopping criterion. The necessary condition for x_k to be a local optimum of the improvement function over the feasible set G_C is

$$0 \in \partial H(x; x) + \sum_{i \in I} \nu_i C_i,$$

The subgradient aggregation provides a good approximation of the subdifferential but it can be too uncertain. Because of that one combines the aggregate subgradient locality measure $\tilde{\beta}_p^k$ and the norm of the current subgradient as follows:

$$\frac{1}{2}\|p_k\|^2 + \tilde{\beta}_p^k < \epsilon_s.$$

By storing a limited number of subgradients this bundle method operates on an approximation of the subdifferential. Indeed, if one applies the subgradient aggregation strategy, one can show global convergence for the bundle method outlined above. A drawback of bundle-type methods is the large number of user defined parameters.

Further explanations and more details of this bundle method can be found in [MN92, Chap. 3].

4.4 Variable Metric Methods

Variable metric methods, also known as quasi-Newton methods, are well established in smooth optimization, because of their good convergence behavior and their reliability. In [LV99], a variable metric method was introduced for convex nonsmooth

unconstrained optimization and global convergence was proven. C. Lemaréchal and C. Sagastizábal considered also convex functions in [LS97] and incorporated additionally bundle strategies. Despite missing convergence results for the nonconvex and nonsmooth case, experiments indicated that variable metric methods are also robust and efficient. This was stated, e.g., in [HUL93, Chap. 8].

A.S. Lewis and M.L. Overton presented a variable metric method in [LO13] applying the BFGS method with an inexact line search. They consider nonsmooth, nonconvex objective functions. To apply the BFGS method on such problems the inexact line search and the termination criteria have to be adapted. The procedure of the BFGS method in general remains unchanged as well as the BFGS update formula. For further information about the BFGS method, see, e.g., [NW06, Chap. 6]. The introduced algorithm terminates if f is not differentiable at the new iterate or if a smooth stationary point is reached. It is considered unlikely that one actually computes a point where f is not differentiable, among others because of numerical rounding errors. Thus, it is reasonable to add an additional termination criterion. The suggestion of the authors is to build up a bundle of gradients G evaluated at nearby points and to solve the quadratic problem

$$\bar{d} = \arg\min\{\|d\| \mid d \in \operatorname{conv} G\}.$$

If $\|\bar{d}\|$ is smaller than a small positive tolerance, the algorithm terminates.

The inexact line search suggested in [LO13] imposes an Armijo condition on the reduction of the function value and a Wolfe condition requiring an algebraic increase in the directional derivative along the line. Contrary to standard line search strategies for nonsmooth optimization, only function values and gradients are required but no subgradients. However, it is assumed that the oracle can detect whether or not f is differentiable at a point x. Under certain stronger assumptions, as, e.g., semi-algebraic functions, termination of the line search can be guaranteed.

To obtain convergence results Lewis and Overton linked the variable metric approach with their gradient sampling method which will be summarized in the subsequent section.

4.5 Gradient Sampling Methods

Gradient sampling methods are some of the latest approaches in nonsmooth optimization. One of the key characteristics of these methods is that no subgradient information is required. The original method was first introduced by J.V. Burke, A.S. Lewis and M.L. Overton in [BLO05], where a locally Lipschitz objective function $f : \mathbb{R}^n \to \mathbb{R}$ is considered, that is continuously differentiable on an open dense subset of $\mathbb{R}^n$. It is also assumed that there is a point $\tilde{x} \in \mathbb{R}^n$ for which $\mathcal{L} = \{x \mid f(x) \leq f(\tilde{x})\}$ is compact. The method is basically constructed as a descent method. The stabilization is controlled by the sampling radius ϵ.

For the purpose of this method Clarke's subdifferential of f at x is represented by

$$\partial_C f(x) = \bigcap_{\epsilon>0} G_\epsilon(x)$$

where the multifunction $G_\epsilon : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is given by

$$G_\epsilon(x) = \text{cl } \text{conv} \nabla f(B_\epsilon(x) \cap D).$$

Since the gradient sampling method is designed to locate a Clarke ϵ-stationary point, i.e., a point that satisfies $0 \in \partial_{C,\epsilon} f(x)$ with the Clarke ϵ-subdifferential defined as

$$\partial_{C,\epsilon} f(x) = \text{cl } \text{conv} \partial_C f(B_\epsilon(x)),$$

the measure of proximity $\rho_\epsilon(x) = \text{dist}(0 \mid G_\epsilon(x))$ was introduced to detect such stationary points.

In the following, the key steps of the gradient sampling method are outlined. At the beginning of the k-th iteration, Clarke's subdifferential $\partial_C f(x^k)$ is approximated by

$$G_k = \text{conv}\{\nabla f(x^{k_0}), \nabla f(x^{k_1}), ..., \nabla f(x^{k_m})\}$$

where $x^k \in \mathbb{R}^n$ is the current iterate, $x^{k_1}, ..., x^{k_m} \in \mathbb{R}^n$ are sampled independently and uniformly from $B_\epsilon(x^k)$ with ϵ the sampling radius, and $G_k \subset G_\epsilon(x^k)$. For all points x^k and x^{k_j}, $j = 1, ..., m$, the function f has to be differentiable, otherwise the algorithm has to be interrupted. The quality of this approximation was analyzed in

[BLO02].

The next step is the computation of a descent direction. Thus, the descent direction is set as $d^k = -g^k/\|g^k\|$ where g^k solves the quadratic problem $\min_{g \in G_k} \|g\|^2$, i.e.,

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

If $\|g^k\| = 0$ holds, the stationarity condition is fulfilled and the algorithm terminates. If this termination criterion is not satisfied, a step length t_k has to be computed.

The last step of each iteration is the update of the current iterate x^k. Due to the construction of the gradient sampling method, the new iterate has to be a differentiable point. Therefore, one accepts $x^{k+1} = x^k + t_k d^k$ only, if it is differentiable, else another point $\hat{x}^k \in B_\epsilon(x^k)$ is chosen randomly in such a way that $x^{k+1} = \hat{x}^k + t_k d^k$ is a differentiable point.

First convergence results of the gradient sampling method can be found in [BLO05]. For a fixed sampling radius it is shown that when f has compact level sets then with probability 1 the algorithm generates a sequence of iterates having at least one cluster point that is Clarke ϵ-stationary. Stronger results can be obtained, if one assumes additionally convexity or smoothness. For a sample radius ϵ that reduces to zero, it was shown that if f has a unique ϵ-stationary point x^*, then the set of all cluster points generated by the gradient sampling algorithm converges to x^*.

Here, the original gradient sampling method by Burke, Lewis, and Overton was illustrated which was introduced in 2005. Since then the method was refined in a variety of ways, e.g., an approach for nonconvex, nonsmooth constrained problems, see [CO12], and an adaptive gradient sampling approach which reduces significantly the number of required gradient evaluations, see [CQ13]. F.E. Curtis and X. Que also combined the adaptive gradient sampling idea with quasi-Newton methods as explained in [CQ15].

5

Towards Gray-Box Optimization

A widespread structure of gradient-based optimization methods is the black-box scheme. It assigns the responsibility of each part of the optimization procedure to either the user or the designer of an optimization method. In [HUL93] one can find a detailed description of the black-box scheme which is sketched in Fig. 5.1. According to this scheme, the designer of an optimization method develops the algorithm without any knowledge of the objective function. All required information of the objective function has to be provided by the user. The information embraces both initial parameters as stopping parameters and values that are needed repeatedly

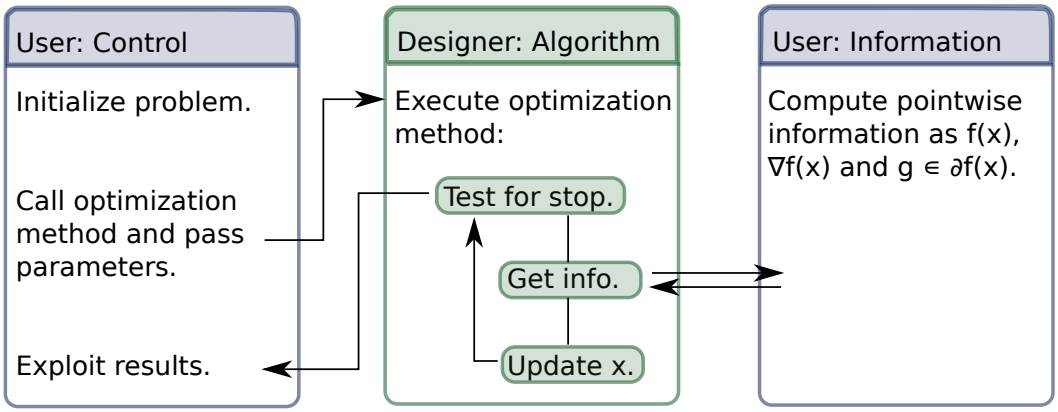


Figure 5.1: Black-box scheme as introduced in [HUL93].

during the optimization process as the function value and derivative information at the current iterate. Additionally, the user has the responsibility for the control and execution of the optimization procedure. Note that the information that are available in this black-box scheme are restricted to be solely pointwise.

All optimization methods introduced in the previous chapter were designed as black-

box methods. The method of the steepest descent trajectory mentioned in Chap. 1 is designed as a black box method as well, see [HUL93]. Nevertheless, the method of the steepest descent trajectory does not only require a single subgradient at each iterate but the full subdifferential. Because of this drawback it was considered to be not implementable.

The algorithm presented in this thesis opens the black box scheme by allowing directional information as the directionally active gradient. In this way, the neighborhood of the current iterate is illuminated and turns gray, as illustrated in Fig. 5.2. These directional components enable the exploitation of the structure of the argument space.

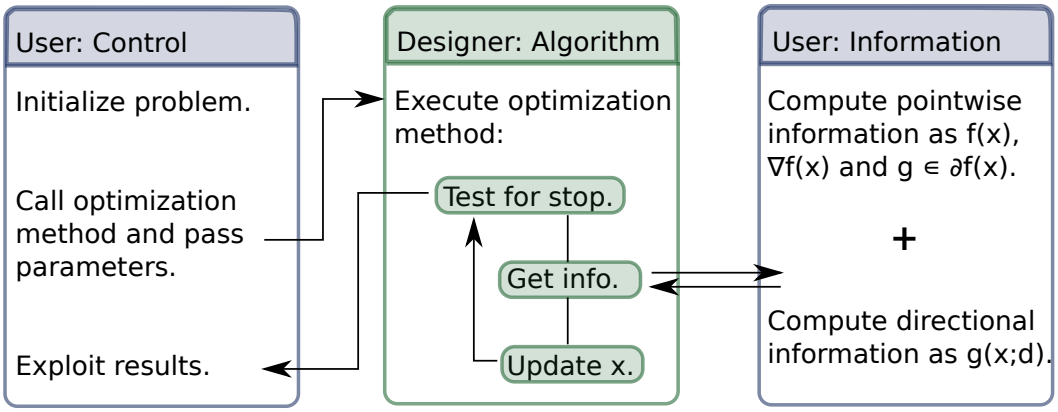


Figure 5.2: Gray-box scheme including directional information

This and the following chapter present the gradient-based minimization algorithm LiPsMin which is the centerpiece of this work. The idea of the algorithm is to minimize a piecewise smooth objective function by generating successively piecewise linearizations at the current iterates and to solve these models by structure exploitation. Therefore, this chapter focuses on the generation of the piecewise linear model and the computation of directional components via the exploitation of the structure caused by the nondifferentiable points. The following chapter presents the overall optimization algorithm including convergence theory.

At the beginning of this chapter, the considered optimization problem is introduced. Using algorithmic differentiation (AD) to compute the piecewise linearization, it is necessary to extend the set of elemental functions by the absolute value function and

to adapt the evaluation procedure appropriately. This is explained in Sec. 5.2. The generation of a piecewise linearization of a piecewise smooth function at a certain base point and the representation of this piecewise linearization in its abs-normal form are summarized in Sec. 5.3. Finally, the computation of directional information via structure exploitation is illuminated in Sec. 5.4.

5.1 Stating the Optimization Problem

The considered nonsmooth optimization problem is of the form

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

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ can be described as the composition of a finite sequence of elemental functions. It is assumed that these elemental functions are either the absolute value function or Lipschitz continuously differentiable on their respective open domain of definition. Such functions will be called composite piecewise differentiable. From the previous assumption it follows that the objective function is a piecewise smooth function as defined in Sec. 3.3. Using the reformulations

$$\begin{aligned} \min(x_1, x_2) &= \frac{1}{2}(x_1 + x_2 - \operatorname{abs}(x_2 - x_1)) \\ \text{and} \quad \max(x_1, x_2) &= \frac{1}{2}(x_1 + x_2 + \operatorname{abs}(x_2 - x_1)) \end{aligned}$$

a quite large range of Lipschitz continuous and piecewise smooth objective functions originated from both real world and academic applications are covered.

5.2 Adapting the Evaluation Procedure

Applying algorithmic differentiation to compute derivative information of the considered nonsmooth objective function, the set of elemental functions Φ has to be extended by the absolute value function. Furthermore, the evaluation scheme has to be adapted appropriately.

The basic idea of AD is the calculation of derivative information of a given function $f : U \subseteq \mathbb{R}^n \to \mathbb{R}$ by the chain rule, see [GW08]. Therefore, the function f has to be a composition of elemental functions $\varphi \in \Phi$ such that f can be given by an evaluation procedure as illustrated in Tab. 5.1. The evaluation procedure consists

v_{i-n}	$=$	x_i	$i = 1 \dots n$
v_i	$=$	$\varphi_i(v_j)_{j \prec i}$	$i = 1 \dots l$
y	$=$	v_l	

Table 5.1: Standard evaluation procedure

of three parts. First, the independent variables are initialized. In the central part, each intermediate variable corresponds to one elemental function, i.e., $v_i = \varphi_i(v_j)_{j \prec i}$ where the precedence relation $j \prec i$ denotes that v_i depends directly on v_j for $j < i$. Finally, the dependent variables are assigned. In the standard approach of AD it is assumed that the elemental functions φ_i are d times continuously differentiable with $1 \leq d \leq \infty$ on their open domain $U_i \subseteq \mathbb{R}^n$.

To evaluate composite piecewise differentiable functions the set of elemental functions Φ is extended by the absolute value function, i.e., the extended elemental set is defined as $\tilde{\Phi} \equiv \Phi \cup \{\text{abs}\}$. Since the absolute value function is not continuously differentiable, the evaluation procedure has to be adapted in that the absolute value function is regarded separately in the evaluation procedure as is illustrated in Tab. 5.2. This adaption was presented in [FWG] by S. Fiege, A. Walther and A. Griewank. For clarity consecutive smooth elemental functions were combined into

v_{i-n}	$=$	x_i	$i = 1 \dots n$
z_i	$=$	$\psi_i(v_j)_{j \prec i}$	$i = 1 \dots s$
σ_i	$=$	$\text{sign}(z_i)$	
v_i	$=$	$\sigma_i z_i = \text{abs}(z_i)$	
y	$=$	$\psi_{s+1}(v_j)_{j \prec s+1}$	

Table 5.2: Reduced adapted evaluation procedure

larger elemental functions ψ_i with $i = 1, ..., s+1$ where $s \in \mathbb{N}$ denotes the number of evaluations of the absolute value function. This yields a reduction of the evaluation procedure. The intermediate value z_i obtained each by larger elemental function ψ_i

with $i = 1, .., s$ represent the arguments of the absolute value function. Hence, they cause the switching in the corresponding derivative values. The vector

$$z = (z_i)_{i=1,..,s} \in \mathbb{R}^s \tag{5.1}$$

is called switching vector and furthermore, it defines the signature vector

$$\sigma = (\sigma_i(x))_{i=1,..,s} \equiv (\text{sign}(z_i(x)))_{i=1,...,s} \in \mathbb{R}^s \tag{5.2}$$

which plays an important role in the structure exploitation.

Example 5.1. We consider the piecewise smooth function

$$f : \mathbb{R}^2 \to \mathbb{R}, f(x_1, x_2) = (x_2^2 - (x_1)_+)_+ \quad \text{with} \quad y_+ \equiv \max(0, y) \tag{5.3}$$

which can be rewritten in terms of the absolute value function as

$$f(x_1, x_2) = \frac{1}{2}(z_2 + |z_2|) \quad \text{with} \quad z_1 = x_1 \quad \text{and} \quad z_2 = x_2^2 - \frac{1}{2}(z_1 + |z_1|).$$

Its reduced adapted evaluation procedure is illustrated in Tab. 5.3.

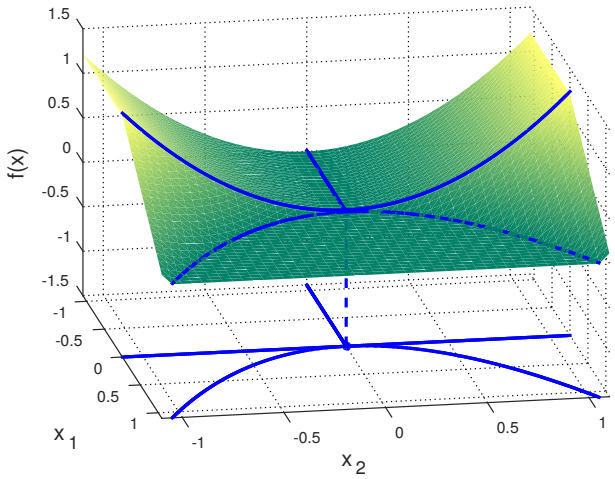


Figure 5.3: Plot of the PS function defined in Ex. 5.1

v_{-1}			$=$	x_1
v_0			$=$	x_2
z_1	$\equiv$	$\Psi_1(v_j)_{j\prec 1}$	$=$	v_{-1}
σ_1			$=$	$\text{sign}(z_1)$
v_1			$=$	$\sigma_1 z_1$
z_2	$\equiv$	$\Psi_2(v_j)_{j\prec 2}$	$=$	$v_0^2 - \frac{1}{2}(z_1 + v_1)$
σ_2			$=$	$\text{sign}(z_2)$
v_2			$=$	$\sigma_2 z_2$
y	$\equiv$	$\Psi_3(v_j)_{j\prec 3}$	$=$	$\frac{1}{2}(z_2 + v_2)$

Table 5.3: Reduced adapted evaluation procedure of Ex. 5.1

5.3 Generating a Piecewise Linearization

To generate a piecewise linearization of a function conforming the adapted evaluation procedure, tangent linearizations of all elemental functions $\varphi \in \tilde{\Phi}$ have to be available. For the elemental functions $\varphi \in \Phi$, these elemental linearization are well-known for a given $\Delta x \in \mathbb{R}^n$ from the standard approach of AD as

$$\begin{aligned}
&\Delta v_i = \Delta v_j \pm \Delta v_k && \text{for } v_i = v_j \pm v_k,\\
&\Delta v_i = v_j * \Delta v_k + v_k * \Delta v_j && \text{for } v_i = v_j * v_k,\\
&\Delta v_i = \varphi'(v_j)_{j\prec i} * \Delta(v_j)_{j\prec i} && \text{for } v_i = \varphi_i(v_j)_{j\prec i} \text{ with } \varphi_i \in \Phi.
\end{aligned}$$

For the absolute value function the tangent approximation

$$\Delta v_i = \text{abs}(z_i + \Delta z_i) - v_i \qquad \text{for } v_i = \text{abs}(z_i) \tag{5.4}$$

was proposed in [Gri13]. This paper by A. Griewank contains among others a detailed analysis of the resulting piecewise linearization which is summarized subsequently. If no absolute value function occurs during the function evaluation, the function $y \equiv f(x)$ is differentiable in a fixed point $x \in \mathbb{R}^n$ and by the chain rule, one obtains

$$\Delta y = \Delta f(x; \Delta x) \equiv \nabla f(x) \Delta x,$$

for a fixed $x \in \mathbb{R}^n$ where $\nabla f(x) \in \mathbb{R}^n$ is the gradient of f. The later equality is no longer true if the absolute value function occurs during the function evaluation. In this case, one obtains for a fixed $x \in \mathbb{R}^n$ the piecewise linear and continuous increment function

$$\Delta y \equiv \Delta f(x; \Delta x) : \mathbb{R}^n \to \mathbb{R}$$

with the argument $\Delta x \in \mathbb{R}^n$, and therefore also the piecewise linearization

$$f_{PL,x}(\Delta x) \equiv f(x) + \Delta f(x; \Delta x) \tag{5.5}$$

of the original PS objective function f.

Example 5.2. The piecewise linearization $f_{PL,x}$ of the function f introduced in Ex. 5.1 evaluated at the base point $\bar{x}$ with the argument $\Delta x = x - \bar{x}$ is given by

$$f_{PL,\bar{x}}(\Delta x) = \frac{1}{2}(z_2 + |z_2|) \tag{5.6}$$

where the switching vector z of $f_{PL,\bar{x}}$ is given by

$$z_1 = \bar{x}_1 + \Delta x_1 \quad \text{and} \quad z_2 = \bar{x}_2^2 + 2\bar{x}_2 \Delta x_2 - \frac{1}{2}(z_1 + |z_1|)\,.$$

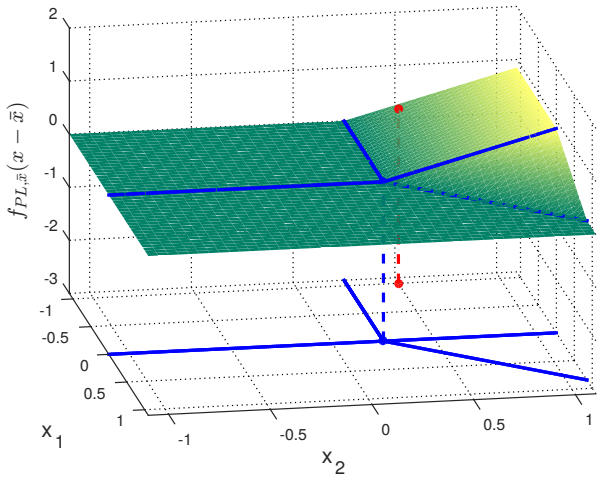


Figure 5.4: Plot of the piecewise linearization with $\bar{x} = (-1, 0.5)$ defined in Ex. 5.2

This piecewise linearization is a second order approximation of the underlying piecewise smooth function as was proven in [Gri13, Prop. 1].

Proposition 5.3. *Suppose $f : \tilde{U} \to \mathbb{R}$ is a function as defined in Sec. 5.1 on some open neighborhood $\tilde{U}$ of a closed convex domain $U \subseteq \mathbb{R}^n$. Then there exists a constant $\gamma \in \mathbb{R}$ such that for all pairs $\bar{x}, x \in U$*

$$\|f(x) - f(\bar{x}) - \Delta f(\bar{x}; x - \bar{x})\| \leq \gamma \|x - \bar{x}\|^2 .$$

Moreover, we have for any pair $\bar{x}, \hat{x} \in U$ and $\Delta x \in \mathbb{R}^n$ and a constant $\tilde{\gamma} \in \mathbb{R}$

$$\frac{\|\Delta f(\bar{x}; \Delta x) - \Delta f(\hat{x}; \Delta x)\|}{1 + \|\Delta x\|} \leq \tilde{\gamma} \|\bar{x} - \hat{x}\| .$$

Proof. See [Gri13, Proposition 1]. □

5.3.1 Representing the Piecewise Linearization in Abs-Normal Form

In [GBRS15] an alternative representation for piecewise linear functions $f_{PL} : \mathbb{R}^n \to \mathbb{R}$ was suggested. It was shown that any piecewise linear function can be expressed using the argument $\Delta x \in \mathbb{R}^n$ and the resulting switching vector $z \in \mathbb{R}^s$ in the abs-normal form given by

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} c_z \\ c_y \end{bmatrix} + \begin{bmatrix} Z & L \\ a^\top & b^\top \end{bmatrix} \begin{bmatrix} \Delta x \\ |z| \end{bmatrix}, \tag{5.7}$$

where $c_z \in \mathbb{R}^s$, $c_y \in \mathbb{R}$, $Z \in \mathbb{R}^{s \times n}$, $L \in \mathbb{R}^{s \times s}$, $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^s$. The matrix L is strictly lower triangular, i.e., each z_i with $i = 1, ..., s$ is assumed to be an affine function of absolute values $|z_j|$ with $j < i$ and the input argument Δx_k for $1 \leq k \leq n$. The structural nilpotency degree of L, i.e., the smallest number $\mu \leq s$ such that

$$L^\mu = 0, \tag{5.8}$$

is called switching depth of f in the given representation. Defining the signature matrix

$$\Sigma \equiv \Sigma(\Delta x) \equiv \operatorname{diag}(\sigma(\Delta x)) \ \in \{-1, 0, 1\}^{s \times s}$$

for the switching vector of the piecewise linearization as defined in Eq. (5.2), one obtains $|z| \equiv \Sigma\, z$ for a fixed $\sigma \in \{-1, 0, 1\}^s$. Solving the first equation of Eq. (5.7) for z by using the signature matrix, the switching vector can be written as

$$z \ = \ (I - L\Sigma)^{-1}(c_z + Z\Delta x). \tag{5.9}$$

Notice that due to the strict triangularity of $L\,\Sigma$ the inverse $(I - L\Sigma)^{-1}$ is well defined and polynomial in the entries of $L\,\Sigma$. Substituting this expression into the last equation of Eq. (5.7), it follows for the function value that

$$f_\sigma(\Delta x) \equiv \gamma_\sigma + g_\sigma^\top \Delta x \tag{5.10}$$

with

$$\gamma_\sigma = c_y + b^\top \Sigma (I - L\Sigma)^{-1} c_z \quad \text{and} \quad g_\sigma^\top = a^\top + b^\top \Sigma (I - L\Sigma)^{-1} Z. \tag{5.11}$$

That is, the gradient evaluation for the piecewise linearization reduces to the solve of a linear system with a triangular matrix. Note that the matrices and vectors c_z, c_y, Z, L, a, and b only depend on the underlying PS function f and the base point $\bar{x}$ in which the PL is evaluated. Therefore, the abs-normal form proves beneficial since it allows an efficient gradient calculation in the optimization algorithm presented in Chap. 6.

Example 5.4. The piecewise linearization $y \equiv f_{PL,\bar{x}}(\Delta x)$ given in Ex. 5.2 can be written in its abs-normal form given by

$$\begin{bmatrix} z_1 \\ z_2 \\ y \end{bmatrix} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2^2 - \frac{1}{2}\bar{x}_1 \\ \frac{1}{2}\left(\bar{x}_2^2 - \frac{1}{2}\bar{x}_1\right) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 2\bar{x}_2 & -\frac{1}{2} & 0 \\ -\frac{1}{4} & \bar{x}_2 & -\frac{1}{4} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ |z_1| \\ |z_2| \end{bmatrix}. \tag{5.12}$$

5.3.2 Realization of the Piecewise Linearization in ADOL-C

The evaluation of the piecewise linearization is realized by the algorithmic differentiation (AD) tool ADOL-C. Further information about this tool can be found in [WG12]. Therefore, the adapted evaluation procedure introduced in Tab. 5.2 and the tangent rule for the absolute value function defined in Eq. (5.5) were included in ADOL-C. All explanatory notes concerning implementations in ADOL-C in this thesis refer to ADOL-C 2.6.3. A description of these drivers was published in [FWKG17] by S. Fiege, A. Walther, K. Kulshreshtha, and A. Griewank.

To generate the corresponding piecewise linearization of a composite piecewise differentiable function f at a fixed point $x \in \mathbb{R}^n$, the driver

```
zos_pl_forward(tag,1,n,1,x,y,z);
```

with the argument x, provides the switching vector $z \in \mathbb{R}^s$ according to Tab. 5.2 and the function value $y = f(x)$. The given `tag` value is used to identify the trace which stores the function evaluation and allows the repeatedly reevaluation of the function and its gradient for different input arguments. Here, the abbreviation zos stands for **z**ero-**o**rder-**s**calar signaling that only a function evaluation and no derivative calculation is performed and the information is propagated forward through the evaluation procedure. One can extract the number of absolute value function evaluations using the call

```
s=get_num_switches(tag);
```

where s is defined in Tab. 5.2. Subsequently, one can use the driver

```
fos_pl_forward(tag, 1, n, x, deltax, y, deltay, z, deltaz);
```

to actually compute the increment $\Delta y = \Delta f(x; \Delta x)$. One also has as output variables the switching vector z and its piecewise linearization Δz. The abbreviation fos stands for first-order-scalar mode.

Beyond that the abs-normal form can be evaluated by ADOL-C. As already mentioned above the components c_z, c_y, a, b, Z and L, of the abs-normal form depend only on the underlying function f and a given base point x in which the PL is evaluated. Hence, ADOL-C provides a driver

```
abs_normal(tag, 1, n, s, x, sigma, y, z, cz, cy, a, b, Z, L);
```

to compute the components of the abs-normal form. This driver requires the routine

```
fos_pl_reverse(tag, 1, n, s, i, res);
```

which applies the adapted handling of the absolute value function in the reverse mode of ADOL-C. The routine returns `res=[Z_i L_i]` for $i = 0, ..., s-1$ and `res=[a^T b^T]` for $i = s$ where $\mathtt{Z}_i$ and $\mathtt{L}_i$ denote the i-th row of the corresponding matrix. Now all components of the abs-normal form are known except c_z and c_y which can be gained by solving the linear system at the base point x for these unknown variables. Note that at the base point x, one has $\Delta x = 0$.

5.4 Computing Directional Information

To obtain directional information of the objective function, the analysis of the polyhedral decomposition caused by the nondifferentiable points is crucial. In Sec. 3.3, it was already mentioned that piecewise smooth functions can be defined via selection functions and that the essentially active index set allows a more exact definition of Clarke's subdifferential. Considering piecewise linear functions $f_{PL} : \mathbb{R}^n \to \mathbb{R}$ in particular, it was determined that every piecewise linear function admits a corresponding polyhedral subdivision of $\mathbb{R}^n$. The exploitation of this decomposition enables the computation of limiting subdifferentials. In the following, the concepts introduced by [Sch12] will be transferred and extended into the setting of this work according to [Gri13, GWFB15].

5.4.1 Description of Piecewise Smooth Functions by Signature Vectors

Adapting the definition of piecewise smooth functions $f : \mathbb{R}^n \to \mathbb{R}$ via selection functions in terms of signature vectors σ as defined in Eq. (5.2), the piecewise smooth function may be written in the form

$$f(x) \in \{f_\sigma(x) \mid \sigma \in \mathcal{E} \subseteq \{-1, 0, 1\}^s\},$$

where the selection functions f_σ are continuously differentiable on neighborhoods of points where they are active, that is, coincide with f. We will assume that all f_σ with $\sigma \in \mathcal{E}$ are essentially active in that their coincidence set $\{f(x) = f_\sigma(x)\}$ are the closures of their interiors, i.e.,

$$\mathcal{E} \equiv \{\sigma(x) \in \{-1, 0, 1\}^s \mid x \in \mathrm{cl}(\mathrm{int}\{x \in \mathbb{R}^n \mid f(x) = f_\sigma(x)\})\}. \tag{5.13}$$

Therewith, one obtains that Clarke's subdifferential can be given by

$$\partial_C f(x) \equiv \mathrm{conv}(\partial_L f(x)) \quad \text{with} \quad \partial_L f(x) \equiv \{\nabla f_\sigma(x) \mid \sigma \in \mathcal{E}\}$$

as was shown in Sec. 3.3.2.

5.4.2 Structure of Decomposed Domain for Piecewise Linear Functions

In the remainder of this chapter, solely piecewise linear functions f_{PL} will be considered. The identification and exploitation of the polyhedral decomposition will be based on the signature vector σ. The decomposition is caused by the nondifferentiable points and consists of at most finitely many convex and relatively open polyhedra P_σ of the form

$$P_\sigma = \{x \in \mathbb{R}^n \mid \sigma = \sigma(x)\}.$$

For any piecewise linear function, it follows by continuity that P_σ must be open but possibly empty if σ is definite, in that all its components are nonzero. For any nonempty P_σ, one has

$$\dim(P_\sigma) \geq n + \|\sigma\|_1 - s = n - s + \sum_{i=1}^{s} |\sigma_i|.$$

When equality holds, the signature σ is called nondegenerate, otherwise critical. In particular degenerated situations, there may be some critical σ that are nevertheless open in that P_σ is open.

The closure of P_σ is given by

$$\overline{P}_\sigma \ \subset \ \{x \in \mathbb{R}^n : f_{PL}(x) = f_\sigma(x)\},$$

with $\sigma = \sigma(x)$ where the selection function f_σ are defined as in Eq. (5.10). Note, that identity must hold in the convex case. In the nonconvex case, f_σ may coincide with f_{PL} at points in other polyhedra $P_{\tilde{\sigma}}$. In fact the coincidence sets may be the union of many polyhedral components. In particular f_σ is essentially active in the sense of Scholtes [Sch12, Chapter 4.1] at all points in $\overline{P}_\sigma$ provided σ is open. To conform with the general concepts of piecewise smooth functions we may restrict f_σ to some open neighborhood of $\overline{P}_\sigma$ such that it cannot be essentially active outside P_σ. Thereby the set of essentially active signature vectors (5.13) can be given by

$$\mathcal{E} = \{\sigma \in \{-1, 0, 1\}^s : P_\sigma \text{ nonempty and open}\}.$$

In [GWFB15] the polyhedral structure and neighboring relations of the polyhedra in terms of the signature vector were studied intensely.

Proposition 5.5 (Polyhedral structure in terms of signature vectors)**.**

(i) The signature vectors are partially ordered by the precedence relation

$$\sigma \preceq \tilde{\sigma} \ :\Longleftrightarrow \ \ \sigma_i^2 \leq \tilde{\sigma}_i \, \sigma_i \quad \textit{for } 1 \leq i \leq s. \tag{5.14}$$

(ii) The closure $\overline{P}_\sigma$ of any P_σ is contained in the extended closure

$$\hat{P}_\sigma \equiv \{x \in \mathbb{R}^n : \sigma(x) \preceq \sigma\} \ \supset \ \overline{P}_\sigma \tag{5.15}$$

with equality holding unless $P_\sigma = \emptyset$.

(iii) The essential signatures $\mathcal{E}$ are exactly the maximal elements amongst all nonempty signatures, i.e.,

$$\mathcal{E} \ni \sigma \prec \tilde{\sigma} \ \ \Longrightarrow \ \ P_{\tilde{\sigma}} = \emptyset \quad \textit{and} \quad \hat{P}_\sigma = \hat{P}_{\tilde{\sigma}},$$

we will call such $\tilde{\sigma}$ extended essential.

(iv) For any two signatures σ and $\tilde{\sigma}$ we have the equivalence

$$\hat{P}_\sigma \subset \hat{P}_{\tilde{\sigma}} \iff \sigma \preceq \tilde{\sigma}.$$

(v) Each polyhedron intersects only the extended closures of its successors

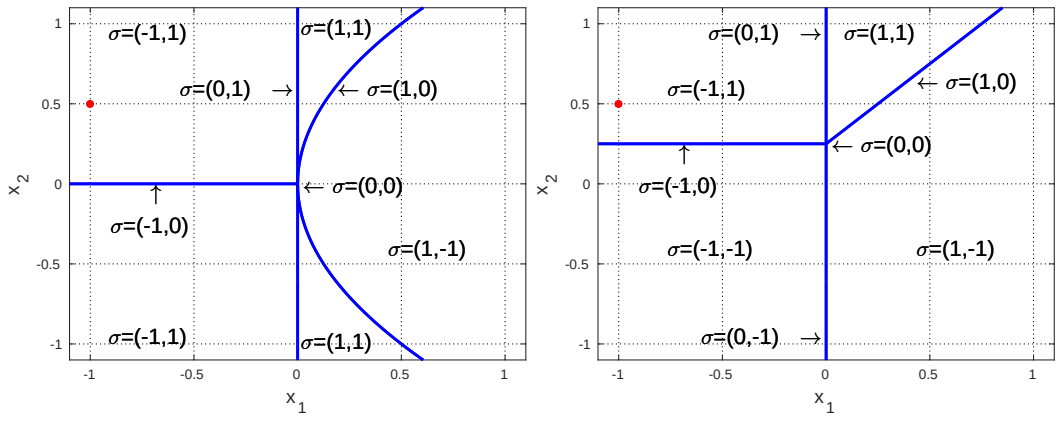
$$P_\sigma \cap \hat{P}_{\tilde{\sigma}} \neq \emptyset \implies \sigma \preceq \tilde{\sigma}.$$

(vi) The closures of the essential polyhedra form a polyhedral decomposition in that

$$\bigcup_{\sigma \in \mathcal{E}} \hat{P}_\sigma = \mathbb{R}^n.$$

Proof. See [GWFB15, Prop. 4.2] □

Example 5.6. Considering the piecewise smooth function defined in Ex. 5.1 and its piecewise linearization evaluated at the base point $\bar{x} = (-1, 0.5)$ in Ex. 5.2, one obtains the decomposition of the argument space as illustrated in Fig. 5.5. Note that the signature vectors of the piecewise smooth function do not coincide with the signature vectors of the piecewise linear function, since the underlying switching vectors differ. Hence, the decompositions of the two functions also differ, see, e.g., the number of polyhedra and the occurring signature vectors.



(a) Decomposed domain of PS function (b) Decomposed domain of PL function

Figure 5.5: Comparison of decompositions of the argument space

5.4.3 Evaluating Directionally Active Gradients and Signature Vectors

The ability to compute limiting gradients $\nabla f_\sigma(x) \in \partial_C f(x)$ of essentially active selection functions f_σ at each point $x \in \mathbb{R}^n$ whether x is differentiable or not, is one of the key ingredients of the algorithm LiPsMin. Whenever $x \in \mathbb{R}^n$ is a nondifferentiable point, this proves to be difficult, since $\sigma \equiv \sigma(x)$ is no guaranteed element of the set $\mathcal{E}$ of essentially active signature vectors. Therefore, the directionally active signature vector $\sigma \equiv \sigma(x; \Delta x)$ and the directionally active gradient g were defined in [Gri13] and the later one is given by

$$g \equiv g(x; \Delta x) \in \partial_L f(x) \quad \text{such that} \quad f'(x; \Delta x) = g^\top \Delta x \tag{5.16}$$

and $g(x; \Delta x)$ equals the gradient $\nabla f_\sigma(x)$ of an essentially active selection function f_σ that coincides with f on a nonempty and open P_σ such that $x, x + \tau \Delta x \in \overline{P}_\sigma$ with $\tau > 0$ arbitrary small. The evaluation of these components in ADOL-C is pointed out in the following subsection.

Besides the directionally active gradient the directionally active signature vector plays an important role in the algorithm LiPsMin. These signature vectors are unlike those signature vectors defined in Eq. (5.2) necessarily elements of the set $\mathcal{E}$ of essentially active signature vectors. Such directionally active signature vectors $\sigma(x; \Delta x)$ are defined by

$$\sigma(x; \Delta x) = (\sigma_i(x; \Delta x))_{i=1,\ldots,s} \equiv (\text{firstsign}(z_i(x); \nabla z_i^\top(x; \Delta x) E))_{i=1,\ldots,s} \tag{5.17}$$

where $\Delta x \in \mathbb{R}^n$ is a preferred direction provided by the user, $E \in \mathbb{R}^{n \times n}$ is a nonsingular matrix and $\text{firstsign}(z; \nabla z^\top(x; \Delta x) E))$ returns for each component σ_i, $i = 1, ..., s$, the sign of the first nonvanishing entry of the vector $(z_i(x); \nabla z_i^\top(x; \Delta x) E) \in \mathbb{R}^{n+1}$. For the application considered in this thesis the matrix E was chosen as

$$E = [\Delta x, e_1, ..., e_{j^*-1}, e_{j^*+1}, e_n] \quad \text{with} \quad j^* = \text{argmax}_{j=1,\ldots,n} |\Delta x_j|$$

where e_i, $i = 1, ..., n$, are the unit vectors. In [Gri13] more information about the firstsign-function can be found.

With regard to an efficient computation of the directional components, the following

finding comes in useful. Let each p_i with $i = 1, ..., s$ be the index of the first nonvanishing entry of the vector $(z_i(x); \nabla z_i^\top(x; \Delta x)E) \in \mathbb{R}^{n+1}$ and $p \equiv \max\{p_1, ..., p_s\}$. It is quite likely that only $p < n$ directions are required to compute a signature vector $\sigma(x; \Delta x)$ which is essentially active. Although, there are cases when n directions are required, e.g., whenever indefinite signatures are contained in $\mathcal{E}$, the implementation of the directional active gradient and signature vector should provide an option to start with a smaller number of directions and to increase it iteratively if required as was already remarked in [Gri13].

Note that combining this directionally active signature vector $\sigma = \sigma(x; \Delta x) \in \mathcal{E}$ with the abs-normal form introduced in subsection 5.3.1, one can compute the corresponding essential active selection function $f_\sigma(x)$ by Eq. (5.10) and its gradient $g_\sigma \equiv \nabla f_\sigma(x)$ by Eq. (5.11). This approach is an efficient and robust way to evaluate limiting subgradients $\nabla f_\sigma(x) \in \partial f_L(x)$.

5.4.4 Realization of Directionally Active Gradients in ADOL-C

ADOL-C provides a driver

directional_active_gradient(tag, n, x, deltax, g, sigxdx);

that returns the directionally active gradient `g` $\equiv g(x; \Delta x)$ and the directionally active signature vector `sigxdx` $\equiv \sigma(x; \Delta x) \in \mathbb{R}^n$ at a given point `x` $\equiv x \in \mathbb{R}^n$ and a direction `deltax` $\equiv \Delta x \in \mathbb{R}^n$. The implementation of this driver was realized in ADOL-C as follows:

```
int directional_active_gradient(tag, n, x, deltax, g, sigxdx){

 keep = 1; by = 1; k = 1; done = 0; j = 0;
 s = get_num_switches(tag);

 E = [deltax];
 max_entry = max_i fabs(deltax[i]);
 max_dk = argmax_i fabs(deltax[i]);

 while((k < p) && (done == 0)){
     fov_pl_forward(tag,1,n,k,x,E,y,deltay,z,deltaz,sigxdx);
```

```
      sum = 0;
      for(i=0;i<s;i++)
          sum += fabs(sigxdx[i]);

      if (sum == s){
          zos_pl_forward(tag, 1, n, keep, x, &y, z);
          fos_pl_sig_reverse(tag, 1, n, s, sigxdx, &by, g);
          done = 1;
      }else{
          if(j==max_dk)
            j++;
          E=[E e_j];
          j++; k++;
      }
  }

  if (done==0){
     \\ Compute g(x; delta x) with full E
  }
}
```

The routine uses those ADOL-C drivers which were introduced in Sec. 5.3.2. Moreover, the routine evaluates the directionally active gradient and the directionally active signature vector simultaneously. In doing so the signature vector is used to reduce the number of required directions as was mentioned in the previous subsection. The routine starts with one direction and checks whether or not the signature is definite. As long as the signature is indefinite further directions are added.

The evaluation of the directionally active signature vector $\sigma(x; \Delta x)$ is realized componentwise in ADOL-C by the following routine:

```
short firstsign(int p, double *z, double* deltaz) {
    int i=0, tmp;
    tmp=((*z)>0.0)?1.0:(((*z)<0.0)?-1.0:0.0);
    while(i<p && tmp==0.0) {
        tmp=(deltaz[i]>0.0)?1.0:((deltaz[i]<0.0)?-1.0:0.0);
        i++;
    }
    return tmp;
}
```

The routine gets `z` $\equiv z_i$ with $i \in \{1, ..., s\}$ and `deltaz` $\equiv \nabla z_i(x; \Delta x)$ as input variables and returns $\sigma_i(x; \Delta x)$.

6

Optimization of Composite Piecewise Differentiable Functions

The main algorithm LiPsMin which was developed and implemented within the scope of this dissertation is presented in detail in this chapter. This includes a description of the solver PLMin of the inner problem, the convergence results, and an overview of possible future extensions. In Fig. 6.1 the idea of LiPsMin is depicted exemplary.

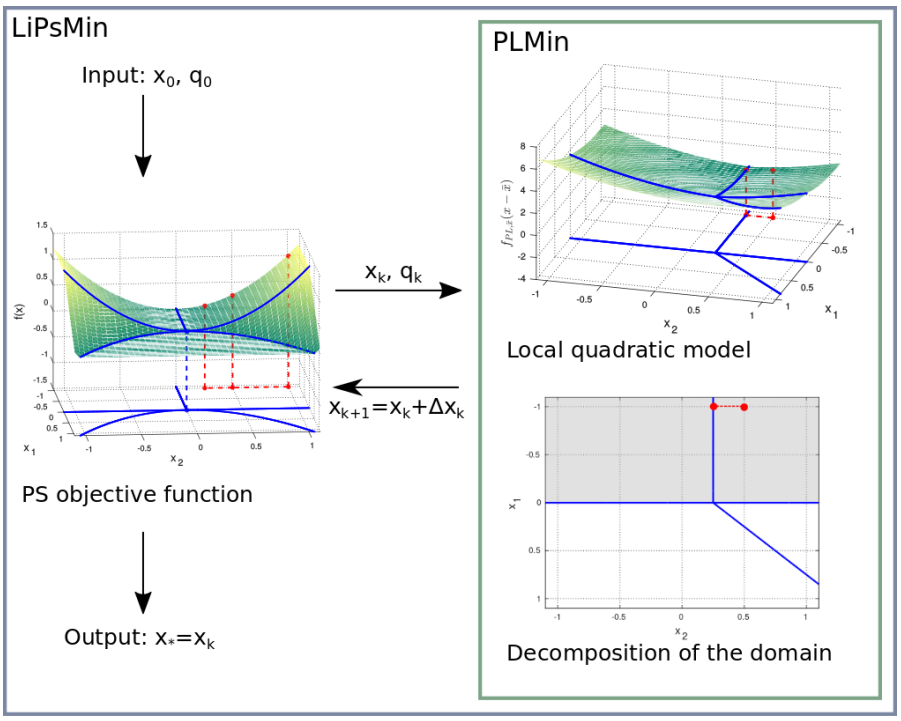


Figure 6.1: Basic idea of algorithm LiPsMin

The algorithm consists of an outer loop that generates successively a piecewise linear model in the current iterate x^k and that calls the inner loop PLMin to solve the local piecewise linear model complemented by a quadratic term. The quadratic overestimation ensures that the local model is bounded below on the one hand, and is required to obtain the desired convergence result towards a stationary point on the other hand. If the solution Δx^k of the local subproblem causes descent in the function values, the iterate is updated, else a nullstep is performed. In both cases, LiPsMin updates the quadratic penalty coefficient q_k.

A detailed description of all steps of LiPsMin is given in this chapter whereby the conceptual algorithm is given as follows:

Algorithm 1 LiPsMin(x, q^0, q^{lb}, κ)

// Precondition: $x \in \mathbb{R}^n$, $\kappa > 0$, $q^0 \geq q^{lb} > 0$.
Set $x^0 = x$.
for $k = 0, 1, 2, \ldots$ **do**

1. Generate a PL model $f_{PL,x^k}(\cdot)$ at the current iterate x^k.
2. Use PLMin(x^k, q^k), see Algo. 2, to solve the overestimated local problem

$$\min_{\Delta x \in \mathbb{R}^n} f_{PL,x^k}(\Delta x) + \frac{1}{2}(1+\kappa) q^k \|\Delta x\|^2.$$

   locally yielding Δx^k.
3. Set $x^{k+1} = x^k + \Delta x^k$ if $f(x^k + \Delta x^k) < f(x^k)$ and $x^{k+1} = x^k$ otherwise.
4. Compute

$$\hat{q}^{k+1} \equiv \hat{q}(x^k, \Delta x^k) \equiv \frac{2|f(x^{k+1}) - f_{PL,x^k}(\Delta x^k)|}{\|\Delta x^k\|^2}$$

   and set

$$q^{k+1} = \max\{\hat{q}^{k+1}, \mu\, q^k + (1-\mu)\, \hat{q}^{k+1}, q^{lb}\}$$

   with $\mu \in [0, 1]$.

end for

How to generate a piecewise linearization $f_{PL,x^k}(\cdot)$ of a piecewise smooth function f at a certain base point x^k was already explained at length in Sec. 5.3. The determination of a Clarke stationary point of the resulting local model composed

of the piecewise linearization $f_{PL,x^k}(\cdot)$ and the quadratic overestimation in step 2 of the algorithm is described in Sec. 6.2, followed by an introduction of the update strategy given in step 4 for the quadratic penalty coefficient in Sec. 6.3. In the algorithmic specification above there was no termination criterion given yet, so that the algorithm generates an infinite sequence $\{x^k\}_{k\in\mathbb{N}}$ that can be examined in the convergence analysis in Sec. 6.4. Nevertheless, a stopping criterion is required for an implementation and therefore, a possible criterion is discussed in Sec. 6.1. In the Sec. 6.5, possible extensions of LiPsMin are presented to reveal future development of the algorithm LiPsMin.

6.1 Stopping Criterion

By now, Algo. 1 generates an infinite sequence $\{x^k\}_{k\in\mathbb{N}}$. For this conceptual algorithm convergence behavior towards a stationary point is studied in Sec. 6.4. Since the purpose of Algo. 1 is the location of a cluster point that is a minimizer of the composite piecewise smooth objective function f, or at least a critical point of f, it is reasonable to use the stopping criterion $\|\Delta x^k\| < \epsilon$ with $\epsilon > 0$. This is approved by the following relations.

Lemma 6.1.

i) If the piecewise smooth function f is locally minimal at x, then the quadratic model $\hat{f}_x$ at x defined as

$$\hat{f}_x(\Delta x) \equiv f_{PL,x}(\Delta x) + \frac{1}{2}\breve{q}\|\Delta x\|$$

with $\breve{q} \equiv (1+\kappa)q$ and $\kappa > 0$ is locally minimal at $\Delta x = 0$ for all $q \geq 0$.

ii) If the quadratic model $\hat{f}_x$ at x is Clarke stationary at $\Delta x = 0$ for some $q \geq 0$, then the piecewise smooth function f is Clarke stationary at x.

Proof. Note that according to [Gri13, Proposition 9] the subdifferential of the piecewise smooth function f at x contains that of the piecewise linearization evaluated

in x at $\Delta x = 0$, i.e.,

$$\partial_C f(x) \supset \partial_C f_{PL,x}(0).$$

We define $h : \mathbb{R}^n \to \mathbb{R}$ as $h(\Delta x) := \frac{\breve{q}}{2}\|\Delta x\|^2$ which is a twice continuously differentiable function with a unique minimizer at $\Delta x = 0$, if $q > 0$. The subdifferential of h is given by $\partial_C h(\Delta x) = \{2\breve{q}\Delta x\}$. Then, the quadratic model at x can be written as $\hat{f}(\Delta x) = f_{PL,x}(\Delta x) + h(\Delta x)$.

i) Let us assume for simplicity that f is locally minimal at x with $f(x) = 0$ and hence $\hat{f}_x(0) = 0$. Suppose that $\hat{f}_x(\cdot)$ is not minimal at 0 for some $q \geq 0$. Then we have for some Δx and $t > 0$

$$\hat{f}_x(t\Delta x) = t g_\sigma^\top \Delta x + o(t) < 0,$$

where we have used the directional differentiability of the piecewise linear model and g_σ is a suitable generalized gradient as defined in Eq. (5.11). Then it follows by the generalized Taylor expansion [Gri13] that for sufficiently small t also

$$f(x + t\Delta x) - f(x) = t g_\sigma^\top \Delta x + o(t) < 0,$$

yielding a contradiction to the minimality of f at x.

ii) If $\hat{f}_x$ generated at x is Clarke stationary in $\Delta x = 0$, then

$$0 \in \partial_C \hat{f}_x(0) = \partial_C (f_{PL,x} + h)(0) \subseteq \partial_C f_{PL,x}(0) + \partial_C h(0).$$

Since $\partial_C h(0) = \{0\}$ one obtains that $0 \in \partial_C f_{PL,x}(0)$. By using the inclusion relation of the subdifferentials noted above this implies that also $0 \in \partial_C f(x)$ and hence f is Clarke stationary in x. □

6.2 Solving the Local Model via PLMin

The solution of the local quadratic model

$$\begin{aligned} \Delta x^k &= \arg\min_{\Delta x \in \mathbb{R}^n} \hat{f}_{x^k}(\Delta x) \\ \text{with} \quad \hat{f}_{x^k}(\Delta x) &\equiv f_{PL,x^k}(\Delta x) + \frac{1}{2}\breve{q}^k \|\Delta x\|^2 \end{aligned} \tag{6.1}$$

and the overestimated quadratic coefficient $\breve{q}^k \equiv (1+\kappa)q^k$ is determined by the inner loop PLMin that exploits the polyhedral structure induced by the nondifferentiabilities of the piecewise linearization. Since we will only consider the k-th iteration of Algo. 1 throughout this section, we use x, Δx and q instead of x^k, Δx^k and q^k.

Example 6.2. We consider the piecewise linear function

$$\begin{aligned} f : \mathbb{R}^2 \to \mathbb{R}, \quad & f(x) := \max\{f_0(x),\ f_{\pm 1}(x),\ f_{\pm 2}(x)\}, \\ \text{with} \quad & f_0(x) := -100,\ f_{\pm 1}(x) := 3x_1 \pm 2x_2,\ f_{\pm 2}(x) := 2x_1 \pm 5x_2 \end{aligned} \tag{6.2}$$

which was already presented in the introduction, see Chap. 1. There it served as an example for the lack of convergence of the steepest descent method combined with an exact line search. In Fig. 6.2 an optimization run generated by PLMin is shown which detects an minimal point in two iterations by exploiting the polyhedral structure.

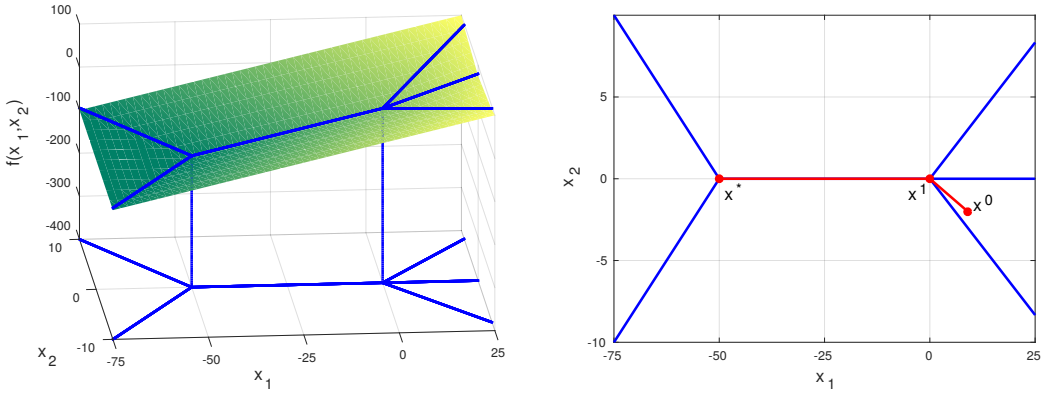


Figure 6.2: Graph of function (6.2) and an optimization run generated by PLMin

The idea of PLMin is to determine a descent trajectory along a path of essential

polyhedra towards a stationary point. For the determination of this trajectory two key problems have to be solved.

At first, a local subproblem has to be defined and solved on each polyhedron along the path of essential polyhedra such that a new segment of the descent trajectory which defines a new iterate is obtained. In [GWFB15] by A. Griewank, A. Walther, S. Fiege, and T. Bosse the *true descent algorithm* was introduced that computes the so called *critical step multiplier* for a given direction in the current iterate such that the new iterate lies on the next nondifferentiability in the given direction. The theoretical results of this method are promising, however the implementation is challenging due to numerical inaccuracy. Therefore, another approach was presented in [FWG] by S. Fiege, A. Walther, and A. Griewank. Instead of computing a critical step multiplier, the solution of a sequence of quadratic subproblems yields the descent trajectory. This sequence of quadratic subproblems is defined in Sec. 6.2.1.

The second key problem is testing the current iterate for stationarity and in case of failure the identification of a proceeding neighboring essential polyhedron that ensures descent. This is realized in both [GWFB15] and [FWG] by similar bundle type methods which benefit from the availability of additional information via structure exploitation. The stationarity test and the determination of a descent direction d are introduced in Sec. 6.2.2. Since both the solution of the local subproblems on essential polyhedra and the computation of the descent direction, require the solution of quadratic problems, practical aspects of the implementation are explained in Sec. 6.2.4.

Convergence results for this algorithm are presented in Sec. 6.2.3.

The above considerations can be summarized as sketched in the following algorithm, where the base point x, the quadratic coefficient q, and the step Δx are used as in Algo. 1. The base point x and the quadratic coefficient q serve as input parameters. The step Δx is the return parameter.

6.2.1 Defining the Sequence of Local Constrained QPs

To define the first constrained quadratic subproblem according to step 1 of Algo. 2, we assume that an initial signature σ^0 corresponding to an essential polyhedron was

Algorithm 2 PLMin(x, q)

// Precondition: $x, \Delta x_j \in \mathbb{R}^n$, $q \geq 0$
Set $\Delta x_0 = 0$.
Identify $\sigma^0 = \sigma(x)$.
for $j = 0, 1, 2, ...$ **do**
1. Determine solution δx_j of local QP (6.1) constrained on the current polyhedron P_{σ^j}.
2. Update $\Delta x_{j+1} = \Delta x_j + \delta x_j$.
3. Compute direction d by ComputeDesDir(Δx_{j+1}, q, $G = \{g_{\sigma^j}\}$), see Algo. 3.
4. If $\|d\| = 0$: STOP.
5. Identify new polyhedron $P_{\sigma^{j+1}}$ using direction d.

end for
Return $\Delta x = \Delta x_{j+1}$.

identified. By constraining problem (6.1) onto the polyhedron P_{σ^0}, one obtains the quadratic subproblem

$$\begin{aligned} \delta x_0 = &\arg\min_{\delta x \in \mathbb{R}^n} f_{\sigma^0}(\delta x) + \frac{1}{2}\breve{q}\|\delta x\|^2, \\ \text{s.t.} \quad & z_i + \nabla z_i^\top \delta x \begin{cases} \leq 0 & \text{if } \sigma_i^0 < 0 \\ \geq 0 & \text{if } \sigma_i^0 > 0 \\ = 0 & \text{if } \sigma_i^0 = 0 \end{cases} \quad \text{for} \quad i = 1, \ldots, s \ , \end{aligned}$$

where $f_{\sigma^0}(\delta x)$ is the selection function corresponding to the essential polyhedron P_{σ^0}, z_i is the i-th component of the switching variable of z, and ∇z_i is the corresponding gradient. All three, the selection function $f_{\sigma^0}(\delta x)$ defined in Eq. (5.10), the switching vector z defined in Eq. (5.9) and its gradient ∇z can be computed via the abs-normal form evaluated in the base point x as explained in Sec. 5.3.1. The given equality constraint is only active in degenerated cases which means that the essential signature is indefinite. In order to solve the j-th subproblem the previous solutions δx_l with $l = 0, ..., j-1$ have to be included as $\Delta x_j = \sum_{l=0}^{j-1} \delta x_l$, such that the relationship between the current essential polyhedron P_{σ^j} and the base point x is

maintained. Hence, one obtains the following general quadratic subproblem

$$\delta x_j = \arg\min_{\delta x \in \mathbb{R}^n} f_{\sigma^j}(\Delta x_j + \delta x) + \frac{1}{2}\breve{q}\,\|\Delta x_j + \delta x\|^2\,, \tag{6.3}$$

$$\text{s.t.} \quad z_i + \nabla z_i^\top (\Delta x_j + \delta x) \begin{cases} \leq 0 & \text{if } \sigma_i^j < 0 \\ \geq 0 & \text{if } \sigma_i^j > 0 \\ = 0 & \text{if } \sigma_i^j = 0 \end{cases} \quad \text{for} \quad i = 1, \ldots, s\,.$$

By solving this sequence of quadratic subproblems for fixed σ^j, one can characterize the points $\tilde{x}$ in the extended closure $\hat{P}_{\sigma^j}$, see Eq. (5.15), that fulfill the system of inequalities. The proximal term added to the piecewise linear local model ensures that the objective function is bounded below. Besides, the objective function is positive definite and quadratic on $\hat{P}_{\sigma^j}$.

Example 6.3. The left graphic of Fig. 6.3 shows the initial iterate x^0 and the corresponding essential polyhedron P_{σ^0} of the optimization run introduced in Exam. 6.2. By solving problem (6.3) one obtains the new iterate as illustrated in the right graphic.

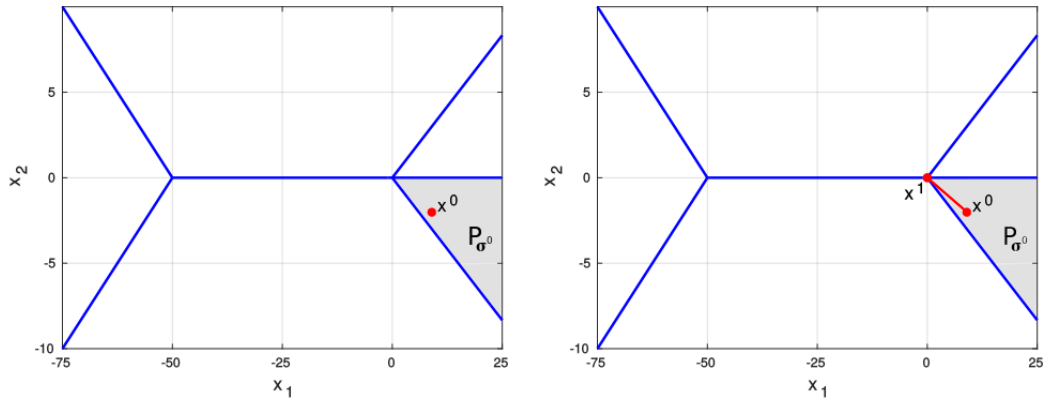


Figure 6.3: Detecting the iterate x^1 by solving the local quadratic problem (6.3).

6.2.2 Stationarity Test and Identification of a Succeeding Polyhedron

The remaining challenge is the determination of a direction d which either fulfills the stationarity test or identifies a new essential polyhedron where the local model

function $\hat{f}_x$ decreases. A Clarke stationary point is detected if $\|d\| = 0$, i.e., $0 \in \partial\hat{f}_x(\Delta x_{j+1})$. If no Clarke stationary point was detected, a descent direction d at Δx_{j+1} and a signature σ^{j+1} have to be found such that $P_{\sigma^{j+1}}$ contains $\Delta x_{j+1} + \tau d$ for small positive τ. For that the following computation of a descent direction can be used:

Algorithm 3 ComputeDesDir(Δx_{j+1}, $\breve{q}$, G)

// Precondition: $\Delta x_{j+1} \in \mathbb{R}^n$, $d \in \mathbb{R}^n$, $\breve{q} \geq 0$, $\emptyset \neq G \subset \partial^L f_{PL,x}(\Delta x_{j+1})$
repeat
  Compute $d = -\mathbf{short}(\breve{q}\Delta x_{j+1}, G)$ as defined in Eq. (6.4).
  Evaluate $g = g(\Delta x_{j+1}; d)$.
  Augment the bundle $G = G \cup \{g\}$.
until $(g + \breve{q}\,\Delta x_{j+1})^\top d \leq -\beta\|d\|^2$
Set $G = \emptyset$.
Return d.

A very similar computation was already proposed in [GWFB15, Algo. 2]. There, it was also proven that the algorithm terminates after finitely many iterations and that it returns a direction d then. If $d = 0$ a stationary point was located. Otherwise, the return vector d is a descent direction. However, for the general case considered here, solely the identification of a polyhedron $P_{\sigma^{j+1}}$ that provides descent compared to the current polyhedron is required. Hence, the additional multiplier $\beta \in (0, 1)$ was introduced to relax the descent condition compared to [GWFB15, Algo. 2].

In this algorithm, the bundle G is a subset of the limiting subdifferential of the piecewise linear function $f_{PL,x}$ at the current iterate Δx_{j+1}. Initially, it contains the gradient g_{σ^j} of the current selection function f_{σ^j}. The direction d is defined as $d = -\text{short}(qx, G)$ with

$$\text{short}(qx, G) = \arg\min \left\{ \|g\| \,\middle|\, g = \sum_{j=1}^{|G|} \lambda_j g_j + qx,\ g_j \in G,\ \lambda_j \geq 0,\ \sum_{j=1}^{|G|} \lambda_j = 1 \right\}. \tag{6.4}$$

Subsequently, the bundle G gets augmented by further directionally active gradients $g(x; d)$ corresponding to neighboring polyhedra computed, e.g., via the abs-normal form as given in Eq. (5.11). A more detailed description of the solution of the

quadratic problem (6.4) is given in the Subsection 6.2.4.

Example 6.4. In Fig. 6.4 the detection of a new essential polyhedron at the current iterate x^1 of Exam. 6.2 is illustrated. This iterate is no stationary point, thus, a subsequent polyhedron has to be identified. In the left graphic, all neighboring essential polyhedra are colored blue. By computing the descent direction d by Algo. 3, one obtains a directions d that points into the essential polyhedron P_{σ^1} that guarantees descent as can be seen in the right graphic.

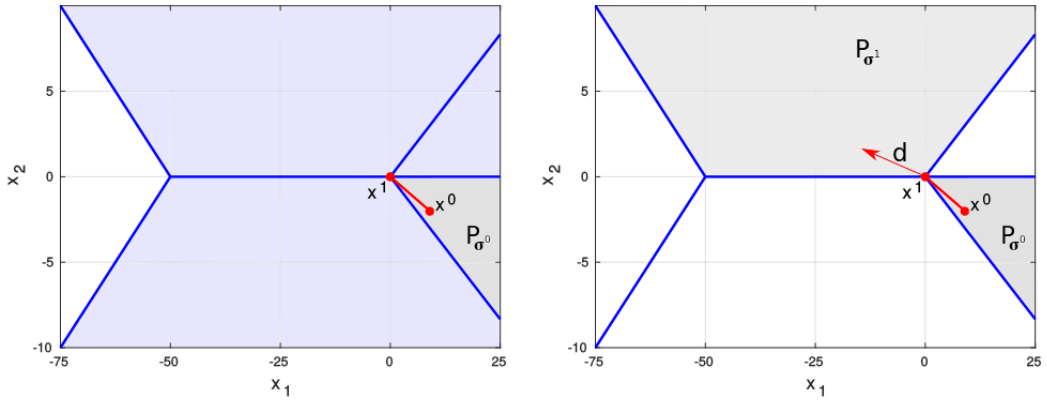


Figure 6.4: Detecting a new essential polyhedron P_{σ^1}

6.2.3 Convergence Results

In [GWFB15] the convergence behavior of a similar version of Algo. 2 was already analyzed and is adjusted for the algorithm presented in this work in this section. Independent of the particular formulation of the algorithm the following lemma was proven in [GWFB15] for the quadratic model $\hat{f}_x$ as defined in (6.1).

Lemma 6.5. *The function $\hat{f}$ attains a global minimum whenever it is bounded below, which must hold if $\breve{q} > 0$.*

Proof. Consider a sequence $\{\Delta x_j\}_{j\in\mathbb{N}} \subset \mathbb{R}^n$ such that

$$-\infty < \inf_{\Delta x \in \mathbb{R}^n} \hat{f}(\Delta x) = \lim_{j\to\infty} \hat{f}(\Delta x_j).$$

Since there are only finitely many polyhedra we may assume without loss of generality that all elements of the infimizing sequence belong to some P_σ so that

$$\hat{f}_x(\Delta x_j) = f_\sigma(\Delta x_j) + g_\sigma^\top \Delta x_j + \frac{\breve{q}}{2}||\Delta x_j||^2$$

with f_σ and g_σ as defined in Eq. (5.10) and (5.11). If $\breve{q} = 0$, the minimization of $\hat{f}_x$ over the closed polyhedron $\overline{P}_\sigma$ can be considered as a linear problem. For linear problems it is well known that feasibility and boundedness imply the existence of an optimal solution which is of course global. If $\breve{q} > 0$ then the Δx_j must be bounded and have a cluster point where $\hat{f}_x$ attains the minimal value. □

Therewith, one obtains the following convergence results for Algo. 2:

Theorem 6.6. *Let $\hat{f}_x : \mathbb{R}^n \to \mathbb{R}$ be the local quadratic model* (6.1) *generated by Algo. 1 with $\breve{q} \geq 0$. Then, Algo. 2 terminates after finitely many iterations. It terminates at a stationary point such that $d = 0$ and returns the increment $\Delta x \in \mathbb{R}^n$ whenever $\hat{f}$ is bounded below, which must hold if $\breve{q} > 0$.*

Proof. If $\breve{q} = 0$, the minimization of $\hat{f}_x$ can be considered as a sequence of linear problems each defined on a essential polyhedron $\overline{P}_\sigma$. Since there are only finitely many polyhedra and Algo. 3 ensures that the successive polyhedron always guarantees descent – if no stationary point was reached yet – a polyhedron has to be reached after finitely many iteration that either contains a stationary point, i.e., $d = 0$, or is unbounded, which is only possible, if $\hat{f}_x$ is unbounded.

If $\breve{q} > 0$, it is guaranteed that $\hat{f}_x$ is bounded below. The minimization of $\hat{f}_x$ can be considered as a sequence of quadratic problems now. Again there are only finitely many polyhedra and Algo. 3 ensures again that the successive polyhedron guarantees descent. In contrast to the case $\breve{q} = 0$ solutions may now be elements of the interior of the current essential polyhedron which has to be checked additionally. Nevertheless, after finitely many iteration a polyhedron has to be reached that contains a stationary point, i.e., $d = 0$. □

6.2.4 Practical Aspects of Solving the Quadratic Subproblems

For the solution of the sequence of quadratic subproblems (6.3) as well as for the computation of the descent direction d by Eq. (6.4) the open source software package qpOASES is applied. A detailed description of the implemented parametric active-set method can be found in [FKP+14]. It detects minimal points of convex quadratic problems and critical points of nonconvex problems, respectively, of the form

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top H x + g^\top x, \quad \text{s.t.} \quad a^l \leq Ax \leq a^u$$

where $H \in \mathbb{R}^{n \times n}$ is a symmetric Hessian matrix, $g \in \mathbb{R}^n$ is a gradient vector, $A \in \mathbb{R}^{m \times n}$ is a constraint matrix, and $a^l, a^u \in \mathbb{R}^m$ are constraint bound vectors. By solving a quadratic problem by qpOASES one obtains additionally the information whether the current polyhedron P_{σ^j} is empty and otherwise whether the function f_{σ^j} is bounded below on the polyhedron, or not.

6.3 Update Strategy for the Penalty Coefficient q

In [Gri13] a first strategy how to compute and update the quadratic coefficient $q^k > 0$ of the quadratic model (6.1) was proposed. However, the strategy only allowed the coefficient to grow. To improve the convergence behavior of Algo. 1 the strategy was adapted in [FWG] such that the estimate q^k can be reduced whenever things are going well.

The development of the update strategy of the *quadratic penalty coefficient* is closely related with the convergence behavior of Algo. 1. Therefore, it is assumed that our composite piecewise differentiable objective function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as described in Sec. 5.1. Additionally, f is assumed to be bounded below and to have a bounded level set $\mathcal{N}_0 \equiv \{x \in \mathbb{R}^n : \ f(x) \leq f(x^0)\}$ with x^0 the starting point of the generated sequence of iterates. Hence, the level set is compact. Furthermore, the objective function f is supposed to satisfy the assumptions concerning the representation of f as an evaluation procedure in Sec. 5.2 on an open neighborhood $\tilde{\mathcal{N}}_0$ of $\mathcal{N}_0$. In the following, the quantities x, Δx, and $\hat{q}$ denote the continuous quantities, while the elements of sequences are marked with a superscript index.

Prop. 5.3 which was first proposed in [Gri13] proves that the piecewise linearization $f_{PL,\hat{x}}(\Delta x)$ as defined in Eq. (5.5) yields a second order approximation of the underlying function $f(x)$. Therewith, it holds

$$\begin{aligned} f(x+\Delta x) &= f(x) + \Delta f(x;\Delta x) + \mathcal{O}(\|\Delta x\|^2) \\ &\leq f(x) + \Delta f(x;\Delta x) + c\|\Delta x\|^2 \end{aligned} \tag{6.5}$$

with the coefficient $c \in \mathbb{R}$. Subsequently, this coefficient is set as $c \equiv \frac{1}{2}\tilde{q}$ whereby the coefficient $\tilde{q}(x;\Delta x)$ can be computed for certain x and Δx. However, it is possible that $\tilde{q}(x;\Delta x)$ is negative and thus, the local quadratic model is not bounded below. Therefore, the coefficient $\hat{q}(x;\Delta x)$ is chosen as

$$\hat{q}(x;\Delta x) \equiv |\tilde{q}(x;\Delta x)| = \frac{2|f(x+\Delta x) - f(x) - \Delta f(x;\Delta x)|}{||\Delta x||^2}. \tag{6.6}$$

By doing so, one obtains from Eq. (6.5) for all descent directions Δx the estimate

$$f(x+\Delta x) - f(x) \leq \Delta f(x;\Delta x) + \frac{1}{2}\hat{q}(x;\Delta x)\|\Delta x\|^2 \leq 0. \tag{6.7}$$

In Proposition 5.3 it was proven as well that there exists a monotonic mapping $\bar{q}(\delta) : [0,\infty) \to [0,\infty)$ such that for all $x \in \mathcal{N}_0$ and $\Delta x \in \mathbb{R}^n$

$$\frac{2|f(x+\Delta x) - f(x) - \Delta f(x;\Delta x)|}{||\Delta x||^2} \leq \bar{q}(||\Delta x||) \tag{6.8}$$

under the assumptions of this section. This holds on one hand because if the line segment $[x, x+\Delta x]$ is fully contained in $\tilde{\mathcal{N}_0}$, then the scalar $\bar{q}(\|\Delta x\|)$ denotes the constant of Proposition 5.3. On the other hand those steps $\Delta x \in \mathbb{R}^n$ for which the line segment $[x, x+\Delta x]$ is not fully contained in $\tilde{\mathcal{N}_0}$ must have a certain minimal size, since the base points x are restricted to $\mathcal{N}_0$. Then the denominators in Eq. (6.8) are bounded away from zero so that $\bar{q}(||\Delta x||)$ exists.

Since $\bar{q}$ is a monotonic descending mapping which is bounded below, it converges to some limit $\bar{q}^* \in (0,\infty)$. Nevertheless, $\bar{q}$ will generally not be known, so that it is approximated by estimates, referred to as *quadratic coefficients* throughout. The sequences of iterates $\{x^k\}_{k\in\mathbb{N}}$ with $x^k \in \mathcal{N}_0$ and the corresponding steps $\{\Delta x^k\}_{k\in\mathbb{N}}$ with $\Delta x^k \in \mathbb{R}^n$ are generated by Algo. 1 and the quadratic coefficient is consistently

updated starting from some $q^0 > 0$ according to

$$q^{k+1} = \max\{\hat{q}^{k+1},\ \mu\, q^k + (1-\mu)\,\hat{q}^{k+1},\ q^{lb}\} \tag{6.9}$$

with $\hat{q}^{k+1} := \hat{q}(x^k; \Delta x^k)$, $\mu \in [0,1]$ and $q^{lb} > 0$ is a lower bound. Therewith, one obtains from Eq. (6.6) and Eq. (6.9) the update strategy of the quadratic penalty coefficient as it was already proposed in Algo. 1.

6.4 Convergence Results

Before the convergence behavior of Algo. 1 is finally analyzed at the end of this section, some properties of the sequences $\{x^k\}_{k\in\mathbb{N}}$, $\{\Delta x^k\}_{k\in\mathbb{N}}$, $\{\hat{q}^k\}_{k\in\mathbb{N}}$, and $\{q^k\}_{k\in\mathbb{N}}$ generated by Algo. 1 are proven.

Lemma 6.7. *Under the assumptions that f is a composite piecewise differentiable function which is bounded below and has a bounded level set $\mathcal{N}_0 \equiv \{x \in \mathbb{R}^n :\ f(x) \leq f(x^0)\}$ with x^0 the starting point of the sequence of iterates $\{x^k\}_{k\in\mathbb{N}}$ generated by Algo. 1, one has:*

a) The sequence of steps $\{\Delta x^k\}_{k\in\mathbb{N}}$ exists.

b) The sequences $\{\Delta x^k\}_{k\in\mathbb{N}}$ and $\{\hat{q}^k\}_{k\in\mathbb{N}}$ are uniformly bounded.

c) The sequence $\{q^k\}_{k\in\mathbb{N}}$ is bounded.

Proof. a) By minimizing the supposed upper bound $\Delta f(x^k; \Delta x) + \frac{1}{2} q^k (1+\kappa)\|\Delta x\|^2$ on $f(x^k + \Delta x) - f(x^k)$ at least locally one always obtains a step

$$\Delta x^k \equiv \arg\min_{\Delta x}(\Delta f(x^k; \Delta x) + \frac{1}{2} q^k (1+\kappa)\|\Delta x\|^2).$$

A globally minimizing step Δx^k must exist since $\Delta f(x^k; \Delta x)$ can only decrease linearly so that the positive quadratic term always dominates for large $\|\Delta x\|$. Moreover, Δx^k vanishes only at first order minimal points x^k where $\Delta f(x^k; \Delta x)$ and $f'(x^k; \Delta x)$ have the local minimizer $\Delta x = 0$.

b) It follows from $q^k \geq q^{lb} > 0$ and the continuity of all quantities on the compact set $\mathcal{N}_0$ that the step size $\delta \equiv \|\Delta x\|$ must be uniformly bounded by some $\bar{\delta}$. This

means that the $\hat{q}$ are uniformly bounded by $\bar{q} \equiv \bar{q}(\bar{\delta})$.

c) The sequence $\{q^k\}_{k\in\mathbb{N}}$ is bounded below by q^{lb}. Considering the first two arguments of Eq. (6.9), one obtains that $q^{k+1} = \hat{q}^{k+1}$ and $q^{k+1} > q^k$ if $\hat{q}^{k+1} > \mu\ q^k + (1-\mu)\ \hat{q}^{k+1}$. Respectively, if $\hat{q}^{k+1} \leq \mu\ q^k + (1-\mu)\ \hat{q}^{k+1}$, one obtains $q^{k+1} \geq \hat{q}^{k+1}$ and $q^{k+1} \leq q^k$. This means that the maximal element of the sequence is given by a $\hat{q}^j$ with $j \in \{1, ..., k+1\}$ and thus bounded by $\bar{q}(\|\Delta x^j\|)$. Therefore, the sequence $\{q^k\}_{k\in\mathbb{N}}$ is bounded above. □

The proof of Lemma 6.7 c) gives us the important insight that $q^{k+1} \geq \hat{q}^{k+1}$ holds. With these results the main convergence result of this thesis can be proven.

Theorem 6.8. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a composite piecewise differentiable function which is bounded below and has a bounded level set $\mathcal{N}_0 \equiv \{x \in \mathbb{R}^n :\ f(x) \leq f(x^0)\}$ with x^0 the starting point of the sequence of iterates $\{x^k\}_{k\in\mathbb{N}}$ generated by Algo. 1. Furthermore, f is assumed to be given by an evaluation procedure as defined in Section 5.2.*

Then a cluster point x^* of the infinite sequence $\{x^k\}_{k\in\mathbb{N}}$ generated by Algo. 1 exists. All cluster points of the infinite sequence $\{x^k\}_{k\in\mathbb{N}}$ are Clarke stationary.

Proof. The sequence of steps $\{\Delta x^k\}_{k\in\mathbb{N}}$ is generated by solving the overestimated quadratic problem in step 2 of Algo. 1 of the form

$$\Delta x^k = \arg\min_{\Delta x}(\Delta f(x;\Delta x) + \frac{1}{2}(1+\kappa)q^k\|\Delta x\|^2).$$

Unless x^k satisfies first order optimality conditions the step Δx^k satisfies

$$\Delta f(x^k;\Delta x^k) + \frac{1}{2}(1+\kappa)q^k\|\Delta x^k\|^2 < 0. \tag{6.10}$$

Therewith, one obtains from Eq. (6.7) in the form

$$\begin{aligned} f(x^k+\Delta x^k) - f(x^k) &\leq \Delta f(x^k;\Delta x^k) + \frac{1}{2}\hat{q}^{k+1}(x^k;\Delta x^k)\|\Delta x^k\|^2 \\ &\leq \Delta f(x^k;\Delta x^k) + \frac{1}{2}q^{k+1}\|\Delta x^k\|^2 \end{aligned}$$

where $\hat{q}^{k+1} \leq q^{k+1}$ holds as a result to Eq. (6.9) and from Eq. (6.10) in the form

$$\Delta f(x^k; \Delta x^k) \leq -\frac{1}{2} q^k (1+\kappa) \|\Delta x^k\|^2$$

the following inequality

$$f(x^k + \Delta x^k) - f(x^k) \leq \frac{1}{2} \left[q^{k+1} - (1+\kappa) q^k \right] \|\Delta x^k\|^2. \tag{6.11}$$

Applying the limit superior $\bar{q} = \limsup_{k \to \infty} q^{k+1}$ to this inequality, it can be overestimated as follows

$$f(x^k + \Delta x^k) - f(x^k) \leq \frac{1}{2} \left[\bar{q} - (1+\kappa) q^k \right] \|\Delta x^k\|^2.$$

Considering a subsequence of $\{q^{k_j}\}_{j \in \mathbb{N}}$ converging to the limit superior, it follows that for each $\epsilon > 0$ a $\bar{j} \in \mathbb{N}$ exists such that for all $j \geq \bar{j}$ one obtains $\|\bar{q} - q^{k_j}\| < \epsilon$. Therewith the overestimated local problem provides that the term $\bar{q} - (1+\kappa) q^{k_j} < 0$. Since the objective function f is bounded below on $\mathcal{N}_0$, infinitely many significant descent steps can not be performed and thus $f(x^{k_j} + \Delta x^{k_j}) - f(x^{k_j})$ has to converge to 0 as j tends towards infinity. As a consequence, the right hand side of Eq. (6.11) has to tend towards 0 as well. Therefore, the subsequence $\{\Delta x^{k_j}\}_{j \in \mathbb{N}}$ is a null sequence. Since the level set $\mathcal{N}_0$ is compact, the sequence $\{x^{k_j}\}_{j \in \mathbb{N}}$ has a subsequence that tends to a cluster point x^*. Hence, a cluster point x^* of the sequence $\{x^k\}_{k \in \mathbb{N}}$ exists.

Assume that the subsequence $\{x^{k_j}\}_{j \in \mathbb{N}}$ of $\{x^k\}_{k \in \mathbb{N}}$ converges to a cluster point. As shown above the corresponding sequence of penalty coefficients $\{\Delta x^{k_j}\}_{j \in \mathbb{N}}$ converges to zero if j tends to infinity. Therewith, one can apply Lemma 6.1 at the cluster point x^*, where it was proven that if $\hat{f}_x$ is Clarke stationary at $\Delta x = 0$ for one $q \geq 0$, then the piecewise smooth function f is Clarke stationary in x yielding the assertion. □

6.5 Possible Extensions

While developing Algo. 1 in the scope of this thesis the focus was put on proving the practicality of the algorithm in terms of convergence behavior and designing an

implementable algorithm. In doing so considering the efficiency of the algorithm did not play an important role. Nevertheless, efficiency should be an important aspect in future developments. Therefore, some possible extensions shall be mentioned in the following.

The generation of the piecewise linear model in step 1 of Algo. 1 can be realized more efficient by exploiting sparsity in the underlying abs-normal form defined in Eq. (5.7), especially the sparsity of the matrices Z and L. By using compressing techniques for the abs-normal form, the amount of gradient evaluations can be reduced distinctly.

Algo. 2 can also be improved in terms of efficiency. First, the quadratic problem in step 1 of Algo. 2 can have a huge number of constraints including many inactive constraints. Therefore, it is reasonable to use a quadratic solver that offers appropriate warm start options.

The other possible improvement and certainly most promising improvement is a different computation of the descent direction in step 3 of Algo. 2. In [GW16] new first and second order optimality conditions for piecewise smooth functions were introduced. These optimality conditions distinguish between minima and saddle points. Furthermore, they allow the computation of a descent direction whenever the optimality conditions are violated. Note that they yield a descent direction without combinatorial effort.

6.6 Survey of Previously Published Work

Parts of this thesis were published previously. A brief overview of these papers and summaries of their contents as well as references to the corresponding chapters of this work are given subsequently.

In 2015 the article *On Lipschitz optimization based on gray-box linearization* was presented by A. Griewank, A. Walther, S. Fiege and, T. Bosse, see [GWFB15]. The purpose of this publication was the optimization of composite piecewise differentiable functions by the development of an implementable version of the method of the steepest descent trajectory as described in [HUL93]. The proposed algorithm was already composed of an outer and an inner loop whereby the focus was put on the inner loop, the so-called *true descent algorithm*, that optimizes the local piece-

wise linear model. This algorithm can be considered as a predecessor of Algo. 2. Therefore, the polyhedral structure of the piecewise linearization was studied as presented in Sec. 5.4.2 and a very similar version of the bundle-type stationarity test that is described in Sec. 6.2.2 of this thesis was already presented. Furthermore, convergence results for the true descent method were proven in that paper similar to those results in Sec. 6.2.3.

In 2016 the article *An algorithm for nonsmooth optimization by successive piecewise linearization* was submitted by S. Fiege, A. Walther and, A. Griewank. It is available on www.optimization-online.org. This paper considered in detail the outer loop that optimizes the composite piecewise differentiable function by successive piecewise linearization and proposed Algo. 1 in the form as it is also proposed in this thesis, see the stopping criterion in Sec. 6.1, the update strategy of the quadratic coefficient in Sec. 6.3 and, the overall convergence proof in Sec. 6.4. Additionally, an improved version of the true descent method was presented that solves a sequence of quadratic problems along a path of essential polyhedra. The resulting Algo. 2 is presented in Sec. 6.2.

Finally, the article *Algorithmic differentiation for piecewise smooth functions: A case study for robust optimization* was presented by S. Fiege, A. Walther, K. Kulshreshtha and, A. Griewank, see [FWKG17]. It complements the previous article by giving more information on the realization of Algo. 1 and Algo. 2. In particular, details of the newly developed drivers that were integrated into the AD-tool ADOL-C are described, see Sec. 5.3.2 and Sec. 5.4.4.

7

Numerical Results

The numerical performance of the introduced Algo. 1 named LiPsMin is investigated in this chapter by comparing it with two further state-of-the-art nonsmooth optimization packages. To compare the methods a set of test problems is defined in Sec. 7.1. The considered software tools and their parameter settings are described at the beginning of Sec. 7.2 which deals mainly with the presentation and discussion of the numerical results generated by the software tools.

7.1 Set of Test Problems

In this section, a set of test problems is presented that will be used to test LiPsMin and to compare it with other nonsmooth optimization methods. The test problems were divided into four categories depending on being piecewise linear or piecewise smooth and, convex or nonconvex, respectively. In each subdivision definitions of all test problems can be found, as well as further information concerning those problems. Furthermore, a great number of these test problems are scalable such that the performance of the algorithms can be analyzed in terms of a growing number of optimization parameters and occurring absolute value functions.

7.1.1 Piecewise Linear and Convex Problems

A list of all piecewise linear and convex test problems is given below including an initial point x^0 of the optimization runs. In Tab. 7.1 further properties are presented such as the possible dimensions n and the amount of absolute values s occurring during the function evaluation depending on the dimension n. Additionally, the ratio of the dimension n and the number of absolute value functions s is given in

the column marked by $n : s$. Furthermore, the optimal value f^* of each function is given and for each problem a reference can be found in this table.

No.	Problem	n	s	$n : s$	f^*	Reference
1	Counterexample	2	$2n$	$n < s$	-100	[HUL93]
2	Goffin	50	$n-1$	$s < n$	0	[MN92]
3	MXHILB	any	$2n-1$	$n \leq s$	0	[HMM04]
4	L1HILB	any	n	$n = s$	0	[LV00]
5	Max1	any	$2n-1$	$n \leq s$	0	[MN92]

Table 7.1: List of piecewise linear and convex test problems

1. **Counterexample of HUL**

$$f(x) = \max\{-100,\ 3x_1 \pm 2x_2,\ 2x_1 \pm 5x_2\}$$
$$x^0 = (9,\ -2)$$

2. **Goffin**

$$f(x) = 50 \max_{1 \leq i \leq 50} x_i - \sum_{i=1}^{50} x_i$$
$$x_i^0 = \quad i - 25.5, \quad \text{for all} \qquad i = 1, ..., 50.$$

3. **MXHILB**

$$f(x) = \max_{1 \leq i \leq n} \left| \sum_{j=1}^{n} \frac{x_j}{i+j-1} \right|$$
$$x_i^0 = \quad 1, \quad \text{for all} \qquad i = 1, ..., n.$$

4. **L1HILB**

$$f(x) = \sum_{i=1}^{n} \left| \sum_{j=1}^{n} \frac{x_j}{i+j-1} \right|$$
$$x_i^0 = \quad 1, \quad \text{for all} \qquad i = 1, ..., n.$$

5. **Max1**

$$f(x) = \max_{1 \le i \le n} |x_i|$$
$$x_i^0 = \quad i, \quad \text{for} \qquad i = 1, ..., n \quad .$$

7.1.2 Piecewise Linear and Nonconvex Problems

The solely piecewise linear and nonconvex test problem is given below. In Tab. 7.2 further properties are presented, as explained in Sub. 7.1.1.

No.	Problem	n	s	$n : s$	f^*	Reference
6	2nd Chebyshev-Rosenbrock	any	$2n-1$	$n \le s$	0	[GO12]

Table 7.2: List of piecewise linear and nonconvex test problems

6. **Second Chebyshev-Rosenbrock**

$$f(x) = \frac{1}{4}|x_1 - 1| + \sum_{i=1}^{n-1} |x_{i+1} - 2|x_i| + 1|$$
$$x_i^0 = \quad -0.5, \quad \text{when} \qquad \text{mod}\,(i,2) = 1, \quad i = 1, ..., n \quad \text{and}$$
$$x_i^0 = \quad 0.5, \quad \text{when} \qquad \text{mod}\,(i,2) = 0, \quad i = 1, ..., n.$$

7.1.3 Piecewise Smooth and Convex Problems

A list of all piecewise smooth and convex test problems is given in Tab. 7.3 where also further properties are presented, as explained in Sub. 7.1.1.

No.	Problem	n	s	$n : s$	f^*	Reference
7	MAXQ	any	$n-1$	$s < n$	0	[HMM04]
8	Chained LQ	any	$n-1$	$s < n$	$-(n-1)2^{1/2}$	[HMM04]
9	Chained CB3 I	any	$2(n-1)$	$n \le s$	$2(n-1)$	[HMM04]
10	Chained CB3 II	any	2	$s \le n$	$2(n-1)$	[HMM04]
11	MAXQUAD	10	455	$n < s$	-0.8414083	[LV00]

Table 7.3: List of piecewise smooth and convex test problems

7. **MAXQ**

$$
\begin{aligned}
f(x) &= \max_{1\le i\le n} x_i^2 \\
x_i^0 &= \quad i, \quad \text{for} \qquad i = 1, ..., n/2 \qquad \text{and} \\
x_i^0 &= -i, \quad \text{for} \qquad i = n/2+1, ..., n.
\end{aligned}
$$

8. **Chained LQ**

$$
\begin{aligned}
f(x) &= \sum_{i=1}^{n-1} \max\left\{-x_i - x_{i+1},\ -x_i - x_{i+1} + (x_i^2 + x_{i+1}^2 - 1)\right\} \\
x_i^0 &= -0.5, \quad \text{for all} \qquad i = 1, ..., n \quad .
\end{aligned}
$$

9. **Chained CB3 I**

$$
\begin{aligned}
f(x) &= \sum_{i=1}^{n-1} \max\left\{x_i^4 + x_{i+1}^2,\ (2-x_i)^2 + (2-x_{i+1})^2,\ 2e^{-x_i+x_{i+1}}\right\} \\
x_i^0 &= 2, \quad \text{for all} \qquad i = 1, ..., n \quad .
\end{aligned}
$$

10. **Chained CB3 II**

$$
\begin{aligned}
f(x) &= \max\left\{\sum_{i=1}^{n-1}\left(x_i^4 + x_{i+1}^2\right),\ \sum_{i=1}^{n-1}\left((2-x_i)^2 + (2-x_{i+1})^2\right),\ \sum_{i=1}^{n-1}\left(2e^{-x_i+x_{i+1}}\right)\right\} \\
x_i^0 &= 2, \quad \text{for all} \qquad i = 1, ..., n \quad .
\end{aligned}
$$

11. **MAXQUAD**

$$
\begin{aligned}
f(x) &= \max_{1\le i\le 5}\left(x^T A^i x - x^T b^i\right) \\
A_{kj}^i &= A_{jk}^i = e^{j/k}\cos(jk)\sin(i), \quad \text{for} \quad j < k, \quad j,k = 1, ..., 10 \\
A_{jj}^i &= \tfrac{j}{10}\left|\sin(i)\right| + \textstyle\sum_{k\ne j}\left|A_{jk}^i\right|, \\
b_j^i &= e^{j/i}\sin(ij), \\
x_i^0 &= 0, \quad \text{for all} \qquad i = 1, ..., 10 \quad .
\end{aligned}
$$

7.1.4 Piecewise Smooth and Nonconvex Problems

Finally, a list of all piecewise smooth and nonconvex test problems is introduced in Tab. 7.4 where further properties are presented as explained in Sub. 7.1.1.

No.	Problem	n	s	$n : s$	f^*	Reference
12	1st Chebyshev-Rosenb.	any	$n-1$	$s < n$	0	[GO12]
13	Number of active faces	any	$n+1$	$n < s$	0	[HMM04]
14	Chained Mifflin 2	10	$n-1$	$s < n$	≈ -6.51	[HMM04]
	Chained Mifflin 2	100	$n-1$	$s < n$	≈ -70.15	[HMM04]
	Chained Mifflin 2	1000	$n-1$	$s < n$	≈ -706.55	[HMM04]
15	Chained Cresent I	any	2	$s \leq n$	0	[HMM04]
16	Chained Cresent II	any	$n-1$	$s \leq n$	0	[HMM04]

Table 7.4: List of piecewise smooth and nonconvex test problems

12. **First Chebyshev-Rosenbrock**

$$f(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} \left| x_{i+1} - 2x_i^2 + 1 \right|$$

$$x_i^0 = -0.5, \quad \text{when} \quad \text{mod}\,(i,2) = 1, \quad i = 1, ..., n \quad \text{and}$$
$$x_i^0 = 0.5, \quad \text{when} \quad \text{mod}\,(i,2) = 0, \quad i = 1, ..., n.$$

13. **Number of active faces**

$$f(x) = \max_{1 \leq i \leq n} \left\{ g\left(-\sum_{j=1}^{n} x_j \right),\ g(x_i) \right\} \quad \text{where} \quad g(y) = \ln\left(|y| + 1\right)$$

$$x_i^0 = 1, \quad \text{for all} \quad i = 1, ..., n \quad .$$

14. **Chained Mifflin 2**

$$f(x) = \sum_{i=1}^{n-1} \left(-x_i + 2(x_i^2 + x_{i+1}^2 - 1) + 1.75 |x_i^2 + x_{i+1}^2 - 1| \right)$$

$$x_i^0 = -1, \quad \text{for all} \quad i = 1, ..., n$$

15. **Chained Cresent I**

$$f(x) = \max\left\{\sum_{i=1}^{n-1}\left(x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1\right),\right.$$
$$\left.\sum_{i=1}^{n-1}\left(-x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right)\right\}$$
$$x_i^0 = -1.5, \quad \text{when} \quad \text{mod}\,(i,2) = 1, \quad i = 1,...,n \quad \text{and}$$
$$x_i^0 = 2, \quad \text{when} \quad \text{mod}\,(i,2) = 0, \quad i = 1,...,n.$$

16. **Chained Cresent II**

$$f(x) = \sum_{i=1}^{n-1}\max\left\{\left(x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1\right),\ \left(-x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right)\right\}$$
$$x_i^0 = -1.5, \quad \text{when} \quad \text{mod}\,(i,2) = 1, \quad i = 1,...,n \quad \text{and}$$
$$x_i^0 = 2, \quad \text{when} \quad \text{mod}\,(i,2) = 0, \quad i = 1,...,n.$$

7.2 Comparison and Discussion of Numerical Results

This section presents the results of a great number of optimization runs which are depict in the same order as they were introduced in the previous section. The results are mainly in tabular form but are complemented by further information and figures. To compare and discuss the different optimization packages, the used software packages are briefly introduced in the following.

7.2.1 Nonsmooth Software Packages and their Parameter Settings

The compared software tools and the respective parameter settings are briefly described subsequently. Although all tools were designed to optimize nonsmooth objective functions of the form considered in this thesis, it is apparent that the three optimization tools differ significantly from each other through, e.g., programming language, available information, etc. Because of that any comparison of the algorithms has to be considered carefully. Nevertheless, comparing the methods can still give an appropriate idea of the performance of LiPsMin.

LiPsMin

The algorithm LiPsMin was introduced extensively in Chap. 5 and Chap. 6. However, certain internal coefficients were just defined theoretical. Their values as used in the implementation of the algorithm are given in Tab. 7.5 as well as the three remaining parameters that have to be defined by the user. To apply LiPsMin the objective function has to be available as an evaluation procedure as explained in Sec. 5.2.

Coefficient of overestimated model, see Eq. (6.1)	κ	0.5
Coefficient of update strategy, see Eq. (6.9)	μ	0.9
Coefficient of termination criterion, see Algo. 3	β	0.5
Initial quadratic coefficient for PL problems	q_0	0.001
Initial quadratic coefficient for PS problems	q_0	0.1
Final accuracy tolerance	`eps`	$1e-8$
Upper bound for number of iterations	maxIter	10000

Table 7.5: Parameter setting of LiPsMin (internal and user-defined parameters)

MPBNGC 2.0

The package MPBNGC is a Fortran implementation described in [Mäk03]. It solves nonsmooth multi-objective optimization problems where the objective function $f : \mathbb{R}^n \to \mathbb{R}^k$ and the constraint function $g : \mathbb{R}^n \to \mathbb{R}^m$ are supposed to be locally Lipschitz continuous. In Sec. 4.3 a very similar proximal bundle method by [MN92] was explained for the case $k = 1$.

To optimize user-defined problems the user has to provide a subroutine that evaluates the functions f and g and single subgradients of these functions. In addition, the user can influence the behavior of the subroutine by choosing several parameters. The amount of parameters is certainly a major drawback of this type of bundle methods, since it is not easy for the user to find out how these parameters influence the behavior of the method and what is the best adjustment for a specific optimization problem. For the comparison of the three optimization methods the parameters were chosen as given in Tab. 7.6.

Line search parameter	`RL`	0.01
Distance measure parameter if f convex	`GAM`	0
Distance measure parameter if f nonconvex	`GAM`	0.5
Final accuracy tolerance	`EPS`	$1e-8$
Upper bound for the size of the bundle	`JMAX`	n
Upper bound for number of iterations	`NITER`	10000
Upper bound of function/gradient evaluations	`NFASG`	10000
Upper bound of function/gradient evaluations per iteration	`JMAX`	1000

Table 7.6: Parameter setting of MPBNGC

HANSO 2.2

The package HANSO is a hybrid algorithm for smooth and nonsmooth, convex and nonconvex objective functions. It combines a BFGS method as introduced in [LO13] based on a weak Wolfe line search and a gradient sampling approach as presented in [BLO05]. However, the gradient sampling method is not used by default since it is mainly required for coherent convergence results. In practical terms the BFGS method works sufficiently well such that the computationally expensive gradient sampling can be neglected.

Analogous to MPBNGC the user of HANSO has to provide a subroutine that evaluates the function f and single, arbitrary subgradients $g \in \partial f(x)$ of the considered objective function. In addition, the user can influence the behavior of the subroutine by choosing several parameters. The parameter setting for the subsequent comparison is given in Tab. 7.7.

Termination tolerance	`options.normtol`	$1e-8$
Evaluation distance	`options.evaldist`	$1e-4$
Maximal number of BFGS iterations	`options.maxit`	10000

Table 7.7: Parameter setting of HANSO

7.2.2 Results of Piecewise Linear and Convex Problems

In the following, the results of the comparison are introduced and discussed. The data obtained from the optimization runs are collected in tables. For each test

problem there is one table that contains the results of all three solvers. The results comprise in each case the detected target value (f^*), the number of function and gradient evaluations ($\#f$ and $\#\nabla f$), and the number of required iterations (Iter). For LiPsMin, the number of gradient evaluations $\#\nabla f$ counts the number of reverse sweeps needed to compute the abs-normal form. Since all problems were solved in only a few seconds and the three solvers were written in different programming languages, a comparison of the computational time was considered as not meaningful.

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	2	-100	3	8	2
MPBNGC	2	-100	7	7	6
HANSO	2	-100	9	9	3

Table 7.8: Results of test problem 1: Counterexample of HUL

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	50	$3.69e-12$	3	100	2
MPBNGC	50	$5.40e-12$	66	66	65
HANSO	50	$2.72e-14$	3178	3178	745

Table 7.9: Results of test problem 2: Goffin

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	$1.24e-8$	4	60	3
	50	$3.40e-8$	10	900	9
	100	$4.37e-8$	20	3800	19
MPBNGC	10	$2.92e-10$	11	11	10
	50	$5.49e-10$	15	15	13
	100	$4.01e-10$	18	18	15
HANSO	10	$1.49e-13$	643	643	259
	50	$2.98e-11$	879	879	382
	100	$2.82e-11$	1233	1233	494

Table 7.10: Results of test problem 3: MAXHILB

The presentation of the results follows the same order as the definition of the test set. Hence, the results of the piecewise linear and convex test problems defined in Sec. 7.1.1 are presented first in Tab. 7.8 - Tab. 7.12.

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	$1.16e-9$	4	33	3
	50	$1.18e-8$	4	153	3
	100	$3.09e-8$	4	303	3
MPBNGC	10	$1.67e-9$	24	24	15
	50	$2.22e-9$	25	25	16
	100	$3.60e-9$	24	24	17
HANSO	10	$6.44e-13$	504	504	226
	50	$2.28e-11$	575	575	303
	100	$5.95e-12$	761	761	403

Table 7.11: Results of test problem 4: L1HILB

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	$6.66e-16$	3	40	2
	50	$4.66e-14$	5	400	4
	100	$3.00e-13$	13	2400	12
MPBNGC	10	$1.19e-13$	29	29	27
	50	$1.05e-12$	123	123	121
	100	$1.86e-9$	176	176	165
HANSO	10	$7.63e-6$	38	38	26
	50	$3.81e-6$	119	119	67
	100	$1.91e-6$	220	220	118

Table 7.12: Results of test problem 5: Max1

As expected LiPsMin uses the additional structure information efficiently to minimize the number of iterations and function value evaluations. However, the number of gradient evaluations is increased in some cases. This is due to fact that the number of gradient evaluations is understood as the number of reverse sweeps needed to compute the abs-normal form (ANF). Therewith, it takes $(s+1)$ gradient evaluations to evaluated one abs-normal form.

7.2.3 Results of Piecewise Linear and Nonconvex Problems

The results of the piecewise linear and nonconvex test problem are given in Tab. 7.13. Non of the three optimization methods succeeds in detecting a local minimizer. Instead, Clarke stationary but nonminimal points are detected. This is not too surpris-

ing since the second Chebyshev-Rosenbrock function has $2^{n-1}-1$ Clarke stationary points but only one minimizer as reported in [GO12]. Beyond that all three optimization methods only promise to detect stationary points in the nonconvex case.

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	0.398438	21	400	20
	50	0.400000	3	200	2
	100	0.400000	3	400	2
MPBNGC	10	0.400390	10000	10000	9807
	50	0.400000	188	188	187
	100	0.350000	251	251	249
HANSO	10	0.399414	1383	1383	514
	50	0.400000	2607	2607	463
	100	0.400001	2278	2278	290

Table 7.13: Results of test problem 6: Second Chebyshev-Rosenbrock

To distinguish minimizers and Clarke stationary points that are no minimal points new first- and second-order optimality conditions were proposed in [GW16]. These optimality conditions are based on the *linear independent kink qualifications* (LIKQ) which is a generalization of LICQ familiar from the smooth, nonlinear optimization theory. In [GW16] it was proven that the second Chebyshev-Rosenbrock functions satisfies LIKQ throughout $\mathbb{R}^n$. Therewith, an adaption of Algo. 2 based on LIKQ can be used to actually minimize the function. Therefore, the original computation of a descent direction can be replaced by a reflection of the signature vector σ on the current polyhedron P_σ into the opposing polyhedron by switching all active signs from 1 to -1 or vice versa.

7.2.4 Results of Piecewise Smooth and Convex Problems

In the following, the results of the piecewise smooth and convex test problems are presented in Tab. 7.14 - Tab. 7.18. A large number of optimization runs detected successfully minimal points. The bundle method MPBNGC stopped once because the number of maximal function and gradient evaluations was reached, see Tab. 7.16. Taking a closer look at the run, it appears that MPBNGC almost located the min-

imizer after a reasonable number of iteration, but could not achieve the demanded accuracy. The quasi-Newton method HANSO failed once, see Tab. 7.18. By enabling the gradient sampling mode, the minimal point could be detected as well as indicated in the additional row marked by (*).

The number of function evaluations and required iterations of all three routines are of comparable order of magnitude. This holds mostly also for the number of gradient evaluations. However, there are cases where the number of required gradient evaluations by LiPsMin exceeds the numbers of the other two routines as can be

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	$2.72e-9$	34	330	33
	50	$1.36e-8$	58	2850	57
	100	$8.09e-9$	129	12800	128
LiPsMin (Sparsity)	10	$2.72e-9$	34	198	33
	50	$1.36e-8$	58	399	57
	100	$8.09e-9$	129	1024	128
MPBNGC	10	$3.45e-9$	126	126	101
	50	$3.79e-9$	577	577	549
	100	$4.46e-9$	1118	1118	1083
HANSO	10	$6.16e-17$	787	787	352
	50	$2.12e-16$	4409	4409	1906
	100	$2.97e-16$	8922	8922	3991

Table 7.14: Results of test problem 7: MAXQ

seen in Tab. 7.14. One reason of the higher number of gradient evaluations is as mentioned before that it strongly depends on the number of absolute functions values occurring during function evaluation, since $s+1$ gradients have to be evaluated to generate an abs-normal form at a time. However, the abs-normal form, especially the matrices Z and L, may be sparse matrices. This sparsity is not exploited yet. Furthermore, if the switching depth ν defined in Eq. (5.8) is minimized by coding the objective function appropriately, the matrices Z and L become even sparser. This effect is demonstrated in Exam. 7.1 by the reformulation of the maximum of a vector and was applied to test problem 7 as presented in Tab. 7.14.

Example 7.1. Let $f : \mathbb{R}^4 \to \mathbb{R}$ be the maximum function. Hence, coding the function by

$$\max v = \max(\max(v_1, v_2), \max(v_3, v_4))$$

yields a switching depth $\nu = 2$ in contrast to the original formulation

$$\max v = \max(\max(v_1, \max(v_2, \max(v_3, v_4))))$$

which yields a switching depth $\nu = 3$. This decreased switching depth leads to a higher sparsity of the corresponding abs-normal forms.

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	−12.727922	15	140	14
	50	−69.296464	15	700	14
	100	−140.00714	15	1400	14
MPBNGC	10	−12.727922	40	40	33
	50	−69.296464	143	143	108
	100	−140.00714	468	468	273
HANSO	10	−12.727922	315	315	100
	50	−69.296464	1238	1238	274
	100	−140.00714	2353	2353	416

Table 7.15: Results of test problem 8: Chained LQ

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	18.000000	12	209	11
	50	98.000000	21	1980	20
	100	198.00000	22	4179	21
MPBNGC	10	18.000000	10000	10000	9999
	50	98.000000	260	260	259
	100	198.00000	2740	2740	2739
HANSO	10	18.000000	608	608	175
	50	98.000000	968	968	149
	100	198.00100	918	918	124

Table 7.16: Results of test problem 9: Chained CB3 I

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	18.000000	91	270	90
	50	98.000000	107	318	106
	100	198.00000	106	315	105
MPBNGC	10	18.000000	46	46	45
	50	98.000000	60	60	59
	100	198.00000	41	41	40
HANSO	10	18.000000	198	198	71
	50	98.000000	202	202	79
	100	198.00000	208	208	69

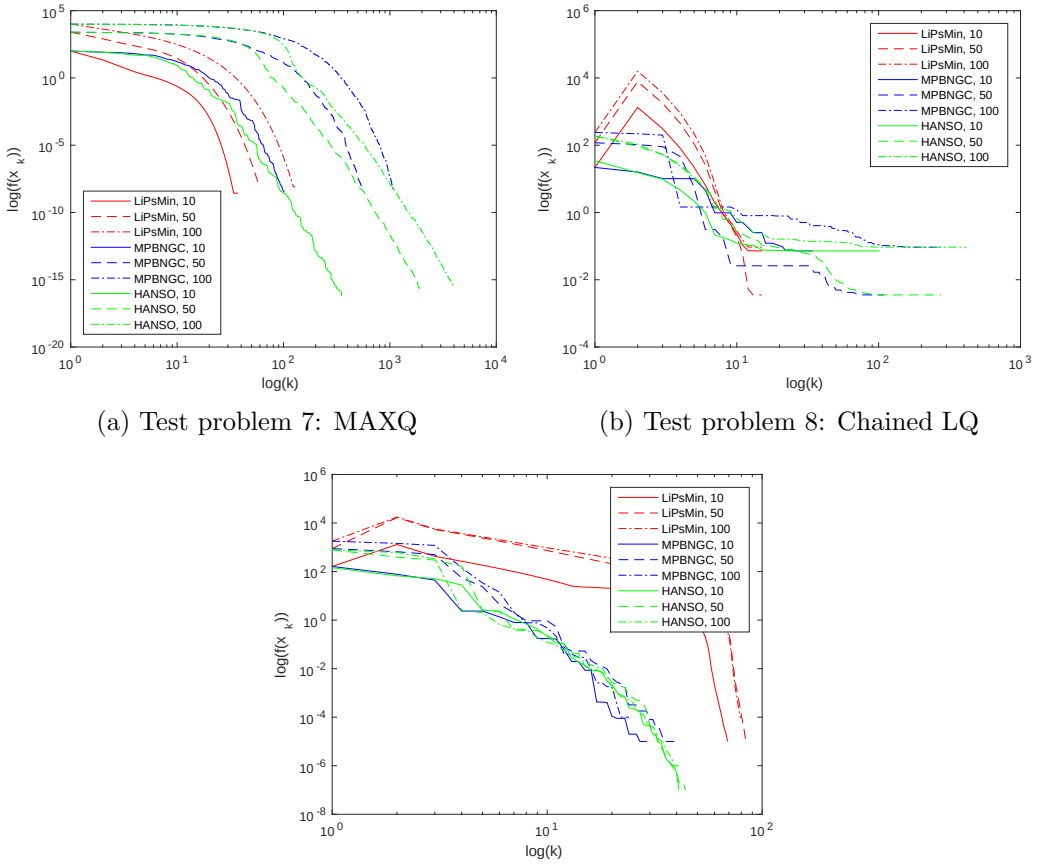
Table 7.17: Results of test problem 10: Chained CB3 II

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	-0.841429	43	210	42
MPBNGC	10	-0.841408	40	40	39
HANSO	10	0	32	32	1
HANSO (*)	10	-0.841397	2943	2943	86

Table 7.18: Results of test problem 11: MAXQUAD

The previous results compared the convergence behavior of the three nonsmooth optimization methods and confirmed well the theoretical considerations derived in Sec. 6.4. However, the rate of convergence was not discussed so far, since there are no theoretical results considering this issue yet. Nevertheless, the rate of convergence is a crucial property of an optimization method. To get a first idea the convergence behavior of three piecewise linear and convex test problems (MAXQ, Chained LQ, Chained CB3 II) is illustrated in Fig. 7.1. These three test problems were chosen since all optimization runs were preformed successfully by each optimization routine. Each figure shows three optimization runs ($n = 10, 50, 100$) for each method (LiPsMin, MPBNGC, HANSO). In particular, the function value $f(x_k)$ is marked on the axis of ordinates while the number of the iterate k is marked on the abscissa. Note that both scales are logarithmically. Considering problems 8 and 9 it strikes that the function value of the first iteration is bigger than the previous one. This is due to a small initial quadratic coefficient q_0 and therewith, a not sufficiently accurate quadratic model. This drawback is adjusted in all further steps by computing appropriate quadratic coefficients based on the second order model

introduced in Sec. 6.3. In total, the results presented in Fig. 7.1 are promising, since the convergence behavior of LiPsMin compares well with the MPBNGC and HANSO.



(a) Test problem 7: MAXQ

(b) Test problem 8: Chained LQ

(c) Test problem 10: Chained CB3 II

Figure 7.1: Comparison of convergence behavior

7.2.5 Results of Piecewise Smooth and Nonconvex Problems

Finally, the results of the piecewise smooth and nonconvex functions are analyzed, see Tab. 7.19 - Tab. 7.23. As can be expected from general theory it turns out that these problems are the most difficult to solve. Nevertheless, only a small amount of optimization runs fails which means that the corresponding optimization runs do not terminate regularly by detecting a Clarke stationary point. The first

Chebyshev-Rosenbrock functions causes difficulties to all three routines as can be seen in Tab. 7.19. The first Chebyshev-Rosenbrock function has only one stationary point which is the global minimizer. No routine is able to locate the minimizer.

As in the piecewise smooth and convex case, the bundle method MPBNGC fails several times because it is not able to gain the necessary accuracy. Additionally, it also tends to terminate with a lower accuracy when successful.

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	0.809757	10001	100000	10000
	50	0.818136	6	250	5
	100	0.818136	6	500	5
MPBNGC	10	0.626470	10000	10000	9786
	50	0.630265	353	353	352
	100	0.006109	2283	2283	1842
HANSO	10	0.817073	18753	18753	100000
	50	0.818136	2654	2654	482
	100	0.818136	5173	5173	757

Table 7.19: Results of test problem 12: First Chebyshev-Rosenbrock

	n	f^*	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	$4.22e-15$	4	66	3
	50	$2.46e-14$	9	816	8
	100	$6.75e-11$	14	2626	13
MPBNGC	10	$7.55e-9$	20	20	15
	50	$4.09e-5$	10000	10000	34
	100	$4.44e-5$	10000	10000	9993
HANSO	10	$8.37e-5$	23	23	11
	50	$2.35e-6$	27	27	11
	100	$1.29e-4$	29	29	11

Table 7.20: Results of test problem 13: Number of active faces

	n	f^*	$\#f$	$\#\nabla f$	Iter
	10	-6.514614	68	670	67
LiPsMin	100	-70.15019	68	6700	67
	10	-6.5146142	206	206	184
MPBNGC	100	-70.149860	100000	100000	9999
	10	-6.5146142	391	391	147
HANSO	100	-70.150188	2600	2600	789

Table 7.21: Results of test problem 14: Chained Mifflin 2

	n	f^*	$\#f$	$\#\nabla f$	Iter
	10	$5.15e-12$	56	110	55
LiPsMin	50	$4.80e-12$	57	112	56
	100	$3.32e-12$	58	114	57
	10	$1.15e-8$	49	49	40
MPBNGC	50	$5.70e-9$	167	167	83
	100	$4.17e-9$	96	96	66
	10	$1.11e-16$	171	171	49
HANSO	50	$1.67e-15$	180	180	52
	100	$1.67e-15$	145	145	46

Table 7.22: Results of test problem 15: Chained Crescent I

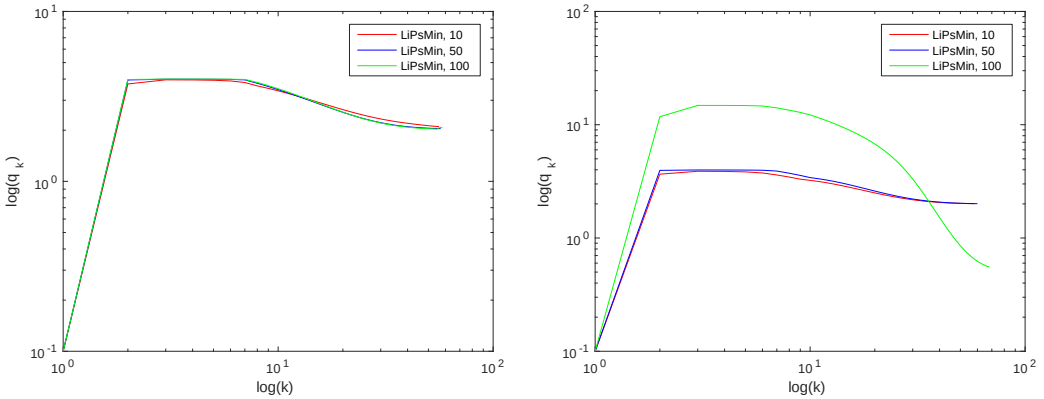
	n	f^*	$\#f$	$\#\nabla f$	Iter
	10	$4.49e-12$	58	570	57
LiPsMin	50	$3.08e-12$	60	2950	59
	100	$3.47e-12$	60	5900	59
	10	$5.07e-9$	196	196	195
MPBNGC	50	$7.10e-9$	519	519	518
	100	$8.01e-9$	733	733	684
	10	$5.22e-15$	626	626	238
HANSO	50	$7.40e-7$	453	453	92
	100	$4.28e-7$	457	457	111

Table 7.23: Results of test problem 16: Chained Crescent II

In Tab. 7.22 and Tab. 7.23 it can be observed how minor changes of the objective function may change the computational effort. Although the overall results are less clear as in the previous section, they are promising since LiPsMin compares well

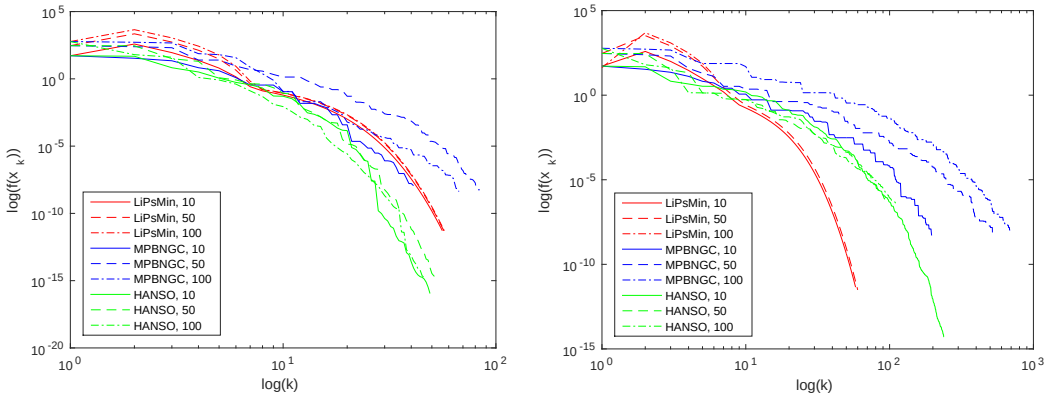
with the other two optimization methods.

As in the piecewise smooth and convex case, the rate of convergence shall be observed exemplary for those functions where all optimization runs located minimizing points successfully. In this case, these are the test problems 15 and 16 (Chained Crescent 1, Chained Crescent 2). Initially, the behavior of the quadratic coefficient q is illustrated in Fig. 7.2. In Lem. 6.7 it was proven that the quadratic coefficient is bounded which is well reflected by the results. Additionally, it can be seen in



(a) Test problem 15: Chained Crescent I (b) Test problem 16: Chained Crescent II

Figure 7.2: Behavior of quadratic penalty coefficient q



(a) Test problem 15: Chained Crescent I (b) Test problem 16: Chained Crescent II

Figure 7.3: Comparison of convergence behavior

Fig. 7.2 and Fig. 7.3 how a small initial quadratic coefficient q_0 causes an increase

of the function value $f(x_1)$ but is immediately corrected by an adjusted quadratic coefficient. Therewith, LiPsMin again converges with a comparable speed as the bundle method MPBNGC and the quasi-Newton method HANSO.

In conclusion one can say that the results of LiPsMin throughout all four subsets of test problems are promising. In terms of robustness, accuracy and convergence behavior it keeps up with the other state-of-the-art optimization routines. Moreover, it is indicated how further exploitation of the polyhedral structure by applying the new first-and second-order optimality conditions proposed in [GW16], and exploitation of sparsity as indicated in Exam. 7.1 can improve the efficiency of LiPsMin.

8 Conclusion

The purpose of this thesis was to develop, implement and examine an algorithm to optimize composite piecewise differentiable objective functions by successive piecewise linearization. Therefore, the main result of this work is the newly developed algorithm LiPsMin. In contrast to numerous conventional nonsmooth algorithms, LiPsMin exploits additional structural information obtained from the successively generated quadratic local model and in particular, from the underlying piecewise linearization. Since the performance of LiPsMin is encouraging, it is important to extract key concepts and ideas, and to reveal future research directions.

8.1 Summary

In this work, the unconstrained, nonconvex, and nonsmooth optimization problem

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

was considered where $f : \mathbb{R}^n \to \mathbb{R}$ is a composite piecewise differentiable function. At the beginning of this thesis, the importance of this class of functions was outlined by sample examples from optimization theory.

Subsequently, fundamental concepts from nonsmooth analysis were introduced concerning convex, Lipschitz continuous, and piecewise smooth functions. These concepts were required to define optimality conditions, to develop LiPsMin, and to finally prove convergence of LiPsMin. Besides, wide-spread ideas and methods from the field of nonsmooth optimization methods were introduce to give an idea of the initial situation of this doctoral study.

The idea of LiPsMin is the optimization of composite piecewise differentiable functions by successive piecewise linearization and by exploitation of the resulting polyhedral structure. Therefore, it was necessary to have the ability to evaluate the piecewise linear local model, to analyze its structure, and to determine how the required information can be obtained. The generation and evaluation of the piecewise linearization was realized by an extension of the AD-tool ADOL-C, i.e., ADOL-C's set of elemental functions was augmented by the absolute value function as well as a tangent approximation of the absolute value functions which ensures that the obtained approximation of the underlying function is of second order. It turned out to be very beneficial to express the piecewise linearization in its abs-normal form, since it allows an efficient computation of all required components. The analysis of the polyhedral decomposed argument space of the piecewise linearization induced by the nondifferentiabilities enabled the determination of directional information as the directionally active gradient. Beyond theoretical considerations, it was outlined how the computation of the afore mentioned components was realized in the implementation.

The final algorithm LiPsMin consists of an outer loop that operates on the original piecewise smooth target function. It is responsible for the generation of the local model which is a composition of the piecewise linearization and a quadratic term that guarantees the lower boundedness of the model. Once the quadratic local model is built, the inner loop PLMin which minimizes the local model by solving a sequence of constrained quadratic subproblems is called. It was proven that PLMin detects a Clarke stationary point after finitely many iterations. Moreover, the convergence of LiPsMin towards a Clarke stationary point was proven which is certainly one of the main results of this thesis.

After developing and analyzing LiPsMin, its numerical performance was examined. Therefore, it was compared with two further state-of-the-art software tools, namely the bundle-type method MPBNGC and the quasi-Newton-type method HANSO. The test set covered piecewise smooth, piecewise linear, convex, and nonconvex problems. The majority of test problems was scalable in its dimension such that the behavior of the optimizations routines for a growing complexity could be observed. The number of required iterations, function and gradient evaluations as well as the finally computed optimal function values were listed for each optimization run. The

results demonstrated that LiPsMin compares well with the other two routines in all compared quantities. Additionally, the rate of convergence was compared for several piecewise smooth problems, and again LiPsMin kept pace with the other two routines.

8.2 Future Research Directions

The algorithm LiPsMin represents a good foundation for further research. The results in Chap. 7 indicate for the considered problems a reasonable rate of convergence. Therefore, it is important to further analyze the convergence behavior of the proposed algorithm such that the observed behavior gets confirmed theoretically.

Since the focus of this thesis was the development of an implementable algorithm, the efficiency of the algorithm was neglected and should be improved. Therefore, three working points were identified. First, exploiting the sparsity properties of the abs-normal form promises to notably reduce the number of gradient evaluations and the amount of required storage. Second, the constrained quadratic subproblems successively solved by PLMin might have a large number of constraints, but they also might have a similar structure which could be exploited by an appropriate QP-solver with proper warm-start options. However, the most promising improvement seems to be a replacement of the bundle-based stationarity test by an approach based on new first- and second-order optimality conditions introduced in [GW16]. This approach promises to test on stationarity without combinatorial effort and yields a descent direction whenever the stationarity test fails.

The previous considerations aimed for an improvement of the algorithm while considering the same optimization problem as in this thesis. Another interesting aspect is an extension of LiPsMin for more general optimization problems. Considering more general objective functions which include, e.g., the Euclidean norm or even jumps is of interest, since there are numerous applications causing such objective functions. However, an extension to these more general functions is not straight forward. Including the Euclidean norm yields an approximation which is no longer of second order and allowing jumps causes objective functions which are no longer Lipschitz continuous. Although these more general functions are much more difficult

to handle, they should be considered due to their great importance. The considered optimization problem can also be generalized by allowing constraints. The degree of difficulty of this purpose depends certainly on the type of constraints such as box-constraints, smooth and nonsmooth equality and inequality constraints. According to the constraints it has to be answered which optimality conditions hold and how the constraints can be integrated in the algorithm.

In summary, it appears that LiPsMin represents already in its current form an algorithm that compares well with other nonsmooth optimization software. The analysis of the structure induced by the nondifferentiabilities proved to be very beneficial and by gaining an even deeper knowledge of this structure further improvement of LiPsMin can be expected for a more efficient and for a more general version of LiPsMin, respectively.

Bibliography

[ANR16] Pierre Apkarian, Dominikus Noll, and Laleh Ravanbod. Nonsmooth bundle trust-region algorithm with applications to robust stability. *Set-Valued and Variational Analysis*, 24(1):115 – 148, 2016.

[BKM14] Adil Bagirov, Napsu Karmitsa, and Marko M. Mäkelä. *Introduction to Nonsmooth Optimization*. Springer, 2014.

[BLO02] James V. Burke, Adrian S. Lewis, and Micheal L. Overton. Approximating subdifferentials by random sampling of gradients. *Mathematics of Operations Research*, 27(3):567 – 584, 2002.

[BLO05] James V. Burke, Adrian S. Lewis, and Micheal L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751 – 779, 2005.

[Cla83] Frank H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, 1983.

[CMO17] Frank E. Curtis, Tim Mitchell, and Micheal L. Overton. A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profils. *Optimization Methods and Software*, 32(1):148 – 181, 2017.

[CO12] Frank E. Curtis and Micheal L. Overton. A sequential quadratic programming method for nonconvex, nonsmooth constrained optimization. *SIAM Journal on Optimization*, 22(2):474 – 500, 2012.

[CQ13] Frank E. Curtis and Xiaocum Que. An adaptive gradient sampling algorithm for nonsmooth optimization. *Optimization Methods and Software*, 28(16):1302 – 1324, 2013.

[CQ15] Frank E. Curtis and Xiaocum Que. A quasi-newton algorithm for nonconvex, nonsmooth optimization with global convergence guarantees. *Mathematical Programming Computation*, 17(4):399 – 428, 2015.

[DV85] Vladimir L. Demyanov and Leonid V. Vasilev. *Nondifferentiable Optimization*. Springer, 1985.

[EG92] Lawrence C. Evans and Ronald F. Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, 1992.

[FKP+14] Hans J. Ferreau, Christian Kirches, Andreas Potschka, Hans G. Bock, and Moritz Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[FWG] Sabrina Fiege, Andrea Walther, and Andreas Griewank. An algorithm for nonsmooth optimization by successive piecewise linearization. submitted, available on www.optimization-online.org.

[FWKG17] Sabrina Fiege, Andrea Walther, Kshitij Kulshreshtha, and Andreas Griewank. Algorithmic differentiation for piecewise smooth functions: A case study for robust optimization. *Optimization Methods and Software*, 2017. http://dx.doi.org/10.1080/10556788.2017.1333613.

[GBRS15] Andreas Griewank, Jens-Uwe Bernt, Manuel Radons, and Tom Streubel. Solving piecewise linear systems in abs-normal form. *Linear Algebra and its Applications*, 471:500–530, 2015.

[GO12] Mert Gurbuzbalaban and Micheal L. Overton. On Nesterov's nonsmooth Chebyshev-Rosenbrock functions. *Nonlinear Analysis: Theory, Methods and Applications*, 75(3):1282 – 1289, 2012.

[Gri13] Andreas Griewank. On stable piecewise linearization and generalized algorithmic differentiation. *Optimization Methods and Software*, 28(6):1139–1178, 2013.

[GW08] Andreas Griewank and Andrea Walther. *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008.

[GW16] Andreas Griewank and Andrea Walther. First- and second-order optimality conditions for piecewise smooth objective functions. *Optimization Methods and Software*, 31(5):904–930, 2016.

[GWFB15] Andreas Griewank, Andrea Walther, Sabrina Fiege, and Torsten Bosse. On Lipschitz optimization based on gray-box piecewise linearization. *Mathematical Programming, Series A*, pages 1–33, 2015.

[HMM04] Marjo Haarala, Kaisa Miettinen, and Marko M. Mäkelä. New limited memory bundle method for large-scale nonsmooth optimization. *Optimization Methods and Software*, 19(6):673–692, 2004.

[HUL93] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms I.* Springer, 1993.

[KB15] Kamil A. Khan and Paul I. Barton. A vector forward mode of automatic differentiation for generalized derivative evaluation. *Optimization Methods and Software*, 30(6):1185–1212, 2015.

[Kel60] J.E. Kelly. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703 – 712, 1960.

[Kiw85] Krzyszof C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization.* Lecture Notes in Mathematics, Springer-Verlag, 1985.

[Kiw90] Krzyszof C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46(1):105–122, 1990.

[KY97] Panos Kouvelis and Gang Yu. *Robust Discrete Optimization and Its Applications.* Springer, 1997.

[Lem78] Claude Lemaréchal. Nonsmooth optimization and descent methods. Research Report RR–78–4, Institut de Recherche en Informatique et Automatique, Le Chesnay, France, 1978.

[LO13] Adrian S. Lewis and Micheal L. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming, Series A*, 141(1):135 – 163, 2013.

[LS97] Claude Lemaréchal and Claudia Sagastizábal. Variable metric bundle methods: From conceptual to implementable forms. *Mathematical Programming*, 76(3):393 – 410, 1997.

[LV98] Ladislav Lukšan and Jan Vlček. A bundle-Newton method for nonsmooth unconstrained minimization. *Mathematical Programming*, 83(1):373 – 391, 1998.

[LV99] Ladislav Lukšan and Jan Vlček. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *Journal of Optimization Theory and Applications*, 102(3):593 – 613, 1999.

[LV00] Ladislav Lukšan and Jan Vlček. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2000.

[Mäk03] Marko M. Mäkelä. Multiobjective proximal bundle method for nonconvex nonsmooth optimization: Fortran subroutine mpbngc 2.0. Reports of the Department of Mathematical Information Technology, Series B, Scientific computing No. B 13/2003, University of Jyväskylä, 2003.

[Mie98] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Springer, 1998.

[Mif82] Robert Mifflin. A modification and an extension of Lemaréchal's algorithm for nonsmooth optimization. *Mathematical Programming Studies*, 17:77 – 90, 1982.

[MN92] Marko M. Mäkelä and Pekka Neittaanmäki. *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control*. World Scientific Publishing Co., 1992.

[Nes05] Yurii Nesterov. Lexicographic differentiation of nonsmooth functions. *Mathematical Programming*, 104(2):669–700, 2005.

[NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

[Roc70] R. Tyrrell Rockafellar. *Convex Analysis*. Princton University Press, 1970.

[Sch89] Helga Schramm. *Eine Kombination von Bundle– und Trust–Region–Verfahren zur Lösung nichtdifferenzierbarer Optimierungsprobleme*. Bayreuther Mathematische Schriften, 1989.

[Sch12] Stefan Scholtes. *An Introduction to Piecewise Differentiable Equations*. Springer, 2012.

[Sho79] Naum Z. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer, 1979. Published by Naukova Dumka in Russian, Kiev, 1979.

[SS05] Claudia Sagastizábal and Mikhail Solodov. An infeasible bundle method for nonsmooth convex constrained optimization without penalty function or a filter. *SIAM Journal on Optimization*, 140(1):146 – 169, 2005.

[WG12] Andrea Walther and Andreas Griewank. Getting Started with ADOL-C. In *Combinatorial Scientific Computing*, pages 181–202. Chapman-Hall CRC Computational Science, 2012.

[YWW13] Gonglin Yuan, Zengxin Wei, and Zhongxing Wang. Gradient trust region algorithm with limited memory BFGS update for nonsmooth convex minimization. *Computational Optimization and Applications*, 54(1):45 – 64, 2013.