SYMBOLIC CONVEX ANALYSIS

by

Chris H Hamilton
B.Sc., Okanagan University College, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the Department
of
Mathematics

SIMON FRASER UNIVERSITY
Spring 2005

APPROVAL

Name: Chris H Hamilton

Degree: Master of Science

Title of thesis: Symbolic Convex Analysis

Examining Committee: Dr. Michael Monagan
Chair

Dr. Jonathan Borwein, Senior Supervisor

Dr. Adrian Lewis, Committee Member

Dr. Rustum Choksi, Committee Member

Dr. Michael McAllister, External Examiner

Date Approved: March 29, 2005

Abstract

Convex optimization is a branch of mathematics dealing with non-linear optimization problems with additional geometric structure. This area has been the focus of considerable research due to the fact that convex optimization problems are scalable and can be efficiently solved by interior-point methods. Over the last ten years or so, convex optimization has found applications in many new areas including control theory, signal processing, communications and networks, circuit design, data analysis and finance. As with any new problem, of key concern is visualization of the problem space in order to develop intuition. In this thesis we develop tools for the visualization of convex functions. An important operation in convex optimization is that of Fenchel conjugation. Earlier research has developed algorithms for the symbolic Fenchel conjugation in one dimension, or many separable dimensions. In this thesis these algorithms are extended to work in the non-separable many dimensional case.

Acknowledgments

I would like to thank Dr. Heinz Bauschke for encouraging me to pursue graduate studies and my supervisor, Dr. Jonathan Borwein, for his invaluable support and guidance, but most of all for his limitless patience in helping me see this work through to completion. This thesis was graciously supported by an NSERC PGS M graduate fellowship, in addition to further financial support from Packeteer Canada and Simon Fraser University. I would also like to thank the people at Packeteer for their support and flexibility in allowing me to return to my studies, especially Paul Archard and Jennifer Nyland.

Contents

Approval ii

Abstract iii

Acknowledgments iv

Contents v

List of Tables vii

List of Figures viii

1 Introduction and Preliminaries 1

1.1 Notation and Convention 2

1.2 Convex Sets and Functions 4

1.3 Closures of Convex Functions 7

1.4 Continuity of Convex Functions 8

1.5 Subgradients and the Subdifferential 12

1.6 The Fenchel Conjugate 17

1.6.1 Concave Functions 22

1.7 Fenchel Duality 23

1.7.1 Examples of Fenchel Duality 26

2 Convex Analysis in One Dimension 29

2.1 A Good Class of Functions 29

2.2 Subdifferentiation 30

2.3 Symbolic Conjugation in One Dimension 32
2.4 Function Inversion 34
2.5 Numerical Methods 38
2.5.1 The Linear-time Legendre Transform 39

3 Convex Analysis in Higher Dimensions 41
3.1 A Good Class of Functions 42
3.2 One dimensional Conjugation With Bounded Parameters 43
3.3 Variable Reordering 43
3.3.1 Region Representation 46
3.3.2 Region Representation to Recursive Representation 47
3.3.3 Region Pivoting 51
3.3.4 Region Pivoting in Two Dimensions 51
3.3.5 Region Swell 53
3.3.6 Boundary Point Problem 53
3.4 Symbolic Conjugation in Higher Dimensions 54
3.5 Numerical Methods 55

4 Applications and Examples 56
4.1 Functionality of the SCAT Package 56
4.2 Ten Classic Examples 58
4.3 Horse Racing Problem 67
4.4 Future Work 70

A Command Listing 72
A.1 Symbolic Convex Analysis Toolkit 72
A.1.1 Introduction 72
A.1.2 Commands 72
A.2 Computational Convex Analysis Toolkit 81
A.2.1 Introduction 81
A.2.2 Commands 81

Bibliography 85

List of Tables

1.1 Some conjugate pairs of one dimensional convex functions 22

List of Figures

1.1 Interpolation characterization of convexity 6

1.2 Some convex subgradients 13

1.3 Vertical intercept interpretation of conjugate 20

1.4 Conjugate relationship for concave functions 23

1.5 Fenchel duality 24

2.1 (a) $f(x)$ and (b) $\partial f(x)$ from Example 2.1 31

2.2 Subdifferential of $\sin x$ on $[\pi, 2\pi]$ 36

3.1 $f(x_1, x_2)$ from Example 3.2 44

3.2 A plan view of $f^*(y_1, y_2)$ from Example 3.2 44

3.3 Pivoting two monotone regions 52

4.1 Plots from Example 4.5 60

4.2 Plot of $(\mathtt{f1}^* + \mathtt{f6}^*)^*$ from Example 4.6 62

4.3 Plot of g8 from Example 4.8 64

4.4 Conjugate pair from Example 4.9 65

Chapter 1

Introduction and Preliminaries

Convex optimization is a branch of mathematics dealing with non-linear optimization problems with additional geometric structure. This area has been the focus of considerable research due to the fact that convex optimization problems are scalable and can be efficiently solved by interior-point methods. Additionally, convex optimization problems are much more prevalent than previously thought as existing problems are constantly being recast in a convex framework.

Over the last ten years or so, convex optimization has found applications in many new areas including control theory, signal processing, communications and networks, circuit design, data analysis and finance. As with any new problem, of key concern is visualization of the problem space in order to help develop intuition. In this thesis we develop and explore tools for the visualization of convex functions and related objects. We provide symbolic functionality where possible and appropriate, and proceed numerically otherwise.

Of critical importance in convex optimization are the operations of Fenchel conjugation and subdifferentiation of convex functions. The algorithms for solving convex optimization problems are inherently numerical in nature, but often closed-form symbolic solutions exist or symbolic computations may be of aid. There exists a wealth of mathematics for assisting the calculation of these operations in closed form, but very little in the way of computer aided tools which take advantage of these techniques. Earlier research has developed algorithms for the manipulation of these objects in one dimension, or many separable dimensions ([2, 3]). In this thesis these tools are extended to work in the non-separable many dimensional case.

In this chapter we explore the basics of convex analysis and develop the theory necessary for a good understanding of the algorithms we will describe in later chapters. We build the subject matter in much the same order as Rockafeller in his classic text [16], but with an emphasis on geometric proofs of results, as in [13]. We also intersperse the fundamentals with more modern results and examples from [4] and [5]. This chapter is intended as a reasonably self-contained introduction to convex analysis up to and including basic results on Fenchel duality.

1.1 Notation and Convention

We begin by discussing the basic geometric and analytic concepts referenced throughout this work. The natural setting for any computer algebra system is $\mathbb{R}^n$, by which we mean an n dimensional vector space over the reals $\mathbb{R}$. However, wherever possible we will present results using an arbitrary Euclidean space $\mathbf{E}$ (a finite dimensional vector space over the reals $\mathbb{R}$ equipped with an inner product $\langle \cdot, \cdot \rangle$), as an abstract coordinate-free representation is often more accessible and elegant.

The *norm* of any point $x \in \mathbf{E}$ is defined as $\|x\| = \sqrt{\langle x, x \rangle}$. The *unit ball* is the set

$$B = \{x \in \mathbf{E} : \|x\| \leq 1\}.$$

The fundamental operations of *set addition* and *set subtraction* for any two sets $C, D \in \mathbf{E}$ are defined as

$$\begin{aligned} C + D &= \{x + y : x \in C, y \in D\}, \quad \text{and} \\ C - D &= \{x - y : x \in C, y \in D\}. \end{aligned}$$

Additionally, for a subset $\Lambda \subset \mathbb{R}$ we define *set scalar multiplication* as

$$\Lambda C = \{\lambda x : \lambda \in \Lambda, x \in C\}.$$

We also represent the standard Cartesian product of two Euclidean spaces $\mathbf{X}$ and $\mathbf{Y}$ as $\mathbf{X} \times \mathbf{Y}$ and define the inner product as $\langle (e, x), (f, y) \rangle = \langle e, f \rangle + \langle x, y \rangle$ for $e, f \in \mathbf{X}$ and $x, y \in \mathbf{Y}$.

We borrow heavily from the language and standard notation of topology. A point x is said to lie in the *interior* of a set $S \subset \mathbf{E}$, denoted by $\operatorname{int} S$, if there is a real $\delta > 0$ such that $N = x + \delta B \subset S$. In this case we say that both N and S are *neighborhoods* of the point x. As an example, the interior of the closed unit ball B is simply the open unit ball $\{x \in \mathbf{E} : \|x\| < 1\}$.

A point $x \in \mathbf{E}$ is the *limit* of a sequence of points $\{x^i\} = x^1, x^2, \ldots$ in $\mathbf{E}$, written $x^j \to x$ as $j \to \infty$ (or $\lim_{j \to \infty} x^j = x$), if $\|x^j - x\| \to 0$. The *closure* S, denoted by cl S, is defined as the set of all limits of all possible sequences in S. The *boundary* of a set S is defined as cl $S \setminus$ int S, and is denoted by bd S. A set S is labelled *open* if $S =$ int S, and *closed* if $S =$ cl S. Basic exercises in set theory show that the *complement* of a set S, written S^c, is open if S is closed (and vice-versa), and that arbitrary unions and finite intersections of open sets remain open.

The interior of a set S may be visualized as the largest open set contained in S, while the closure of a set S is simply the smallest closed set encapsulating S.

We adopt the usual definition and call a map $A : \mathbf{E} \to \mathbf{Y}$ *linear* if all points $x, y \in E$ and all $\lambda, \mu \in \mathbb{R}$ satisfy the equation $A(\lambda x + \mu y) = \lambda Ax + \mu Ay$. The *adjoint* of this map, $A^* : \mathbf{Y} \to \mathbf{E}$, is defined by the constraint

$$\langle A^* y, x \rangle = \langle y, Ax \rangle, \ \forall x \in \mathbf{E}, \ \forall y \in \mathbf{Y}.$$

We also adopt the notation $A^{-1}H$ to denote the *inverse image* of a set H under a mapping A, defined as $A^{-1}H = \{x \in \mathbf{E} : Ax \in H\}$.

In convex analysis it is both natural and convenient to allow functions to take on the value of $+\infty$. For the sake of simplicity we introduce the *extended real numbers*, $\overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$. We further denote the non-negative reals by $\mathbb{R}_+$ and the positive reals by $\mathbb{R}_{++}$.

In allowing functions to take on extended values we are lead to situations in which arithmetic calculations involving $+\infty$ and $-\infty$ must be performed. In dealing with this we adopt the following conventions, used in [4, 16]:

$$\begin{aligned}
&\alpha + \infty = \infty + \alpha = \infty \text{ for } -\infty < \alpha \leq +\infty, \\
&\alpha - \infty = -\infty + \alpha = -\infty \text{ for } -\infty \leq \alpha < +\infty, \\
&\alpha\infty = \infty\alpha = \infty, \quad \alpha(-\infty) = (-\infty)\alpha = -\infty \text{ for } 0 < \alpha \leq \infty, \\
&\alpha\infty = \infty\alpha = -\infty, \quad \alpha(-\infty) = (-\infty)\alpha = \infty \text{ for } -\infty \leq \alpha < 0, \\
&0\infty = \infty 0 = 0(-\infty) = (-\infty)0 = 0, \\
&-(-\infty) = \infty, \quad \inf \emptyset = +\infty, \text{ and } \sup \emptyset = -\infty.
\end{aligned}$$

The troublesome case of $+\infty - \infty$ is generally avoided, but if encountered we adopt the convention of [4] and say $+\infty - \infty = +\infty$, such that any two (possibly empty) sets C and D on $\mathbb{R}$ satisfy the equation $\inf C + \inf D = \inf \{C + D\}$.

1.2 Convex Sets and Functions

Of prime importance in convex optimization is the notion of convexity. We say a set $C \subset \mathbf{E}$ is a *convex set* if all line segments between any two points $x, y \in C$ are themselves contained in the set. In other words, if $(1 - \lambda)x + \lambda y \in C$, for all $x, y \in C$ and for all $\lambda \in [0, 1]$.

Half-spaces are simple but important examples of convex sets. For any non-zero $b \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$, the sets

$$\{x : \langle x, b\rangle \leq \beta\}, \qquad \{x : \langle x, b\rangle \geq \beta\}$$

are called *closed half-spaces*. Similarly, the sets

$$\{x : \langle x, b\rangle < \beta\}, \qquad \{x : \langle x, b\rangle > \beta\}$$

are called *open half-spaces*. All four such sets are plainly non-empty and convex.

We begin with a few basic results regarding set theoretic operations that preserve convexity.

Theorem 1.1 (Intersection of convex sets) ***([16], Theorem 2.1, page 10)*** *The intersection $C = \bigcap C_i$ of an arbitrary collection of convex sets is itself convex.*

Proof: Consider $x, y \in C$. For all i we see that $x, y \in C_i$, and trivially the line segment joining them is as well. Hence C is by definition convex. ■

Theorem 1.2 (Linear images and pre-images of convex sets) ***([16], Theorem 3.4, page 19)*** *Let A be a linear transform from $\mathbb{R}^n$ to $\mathbb{R}^m$. Then AC is a convex set in $\mathbb{R}^m$ for every convex set C in $\mathbb{R}^n$, and $A^{-1}D$ is a convex set in $\mathbb{R}^n$ for every convex set D in $\mathbb{R}^m$.*

Proof: Suppose $x, y \in C$. Since C is convex we know that $(1 - \lambda)x + \lambda y \in C$ for all $\lambda \in [0, 1]$. Due to the linearity of A we also see that $A((1 - \lambda)x) + A(\lambda y) = (1 - \lambda)Ax + \lambda Ay$ is in AC for every $Ax, Ay \in C$. Hence AC is also convex. A similar argument can be used to show that $A^{-1}D$ is convex. ■

The notion of convexity may be extended to real-valued functions but we must first introduce the *epigraph*. The epigraph of a real-valued function defined on a subset $S \subset \mathbf{E}$ $f : S \rightarrow \mathbb{R}$ is denoted by epi f and consists of all points in $\mathbf{E} \times \mathbb{R}$ that lie above the function:

$$\text{epi } f = \{(x, \lambda) \in \mathbf{E} \times \mathbb{R} : x \in S, \lambda \geq f(x)\}.$$

The definition for extended real-valued functions $f : S \subset \mathbf{E} \rightarrow \overline{\mathbb{R}}$ is analogous.

A function $f : \mathbf{E} \rightarrow \overline{\mathbb{R}}$ is said to be a *convex function* if epi f is a convex set in $\mathbf{E} \times \mathbb{R}$. A trivial example of a convex function is the *indicator function* of a convex set. Given a convex set $S \subset \mathbf{E}$, consider the following function $\delta_S : \mathbf{E} \rightarrow \overline{\mathbb{R}}$:

$$\delta_S(x) = \begin{cases} 0, & x \in S \\ +\infty, & x \notin S. \end{cases}$$

From the convexity of S in the space $\mathbf{E}$ it is apparent that epi $\delta_S = S \times \mathbb{R}_+$ is convex in $\mathbf{E} \times \mathbb{R}$.

Stepping outside the language of convex sets, this is equivalent to saying that if the mean value of any two function values is greater than the function value of the mean, then the function is convex. This notion is captured in the following result.

Theorem 1.3 (Interpolation characterization of convexity) *([16], Theorem 4.1, page 25) Consider a function f defined on a set $S \subset \mathbf{E}$, where $f : S \rightarrow \mathbb{R}$. It follows that f is convex if and only if $f((1-\lambda)a + \lambda b) \leq (1-\lambda)f(a) + \lambda f(b)$, for all $a, b \in S$ and $\lambda \in [0, 1]$.*

(In fact, for a proof of the convexity of f we need only show that the given relation holds for any single fixed $\lambda \in (0, 1)$.)

Proof: Suppose that f is convex. By the definition of convexity of a function this is equivalent to saying that epi f is a convex set in $\mathbf{E} \times \mathbb{R}$. Thus we have that all points $(1 - \lambda)f(a) + \lambda f(b)$ are in epi f for any $a, b \in S$ and $\lambda \in [0, 1]$. By the definition of the epigraph, it follows that $(1 - \lambda)f(a) + \lambda f(b) \geq f((1 - \lambda)a + \lambda b)$.

Suppose f is not convex. Then there exists two points $a, b \in$ epi f and some point in between them $c = (1 - \lambda)a + \lambda b \notin$ epi f, for $\lambda \in (0, 1)$. Since $a \in$ epi f then $a = [x_a, r_a]$, where $r_a \geq f(x_a)$, and similarly for b. Since c is outside of epi f then $r_c < f(x_c)$. Thus we see that $(1 - \lambda)f(x_a) + \lambda f(x_b) = (1 - \lambda)r_a + \lambda r_b = r_c < f(x_c) = f((1 - \lambda)x_a + \lambda x_b)$. ■

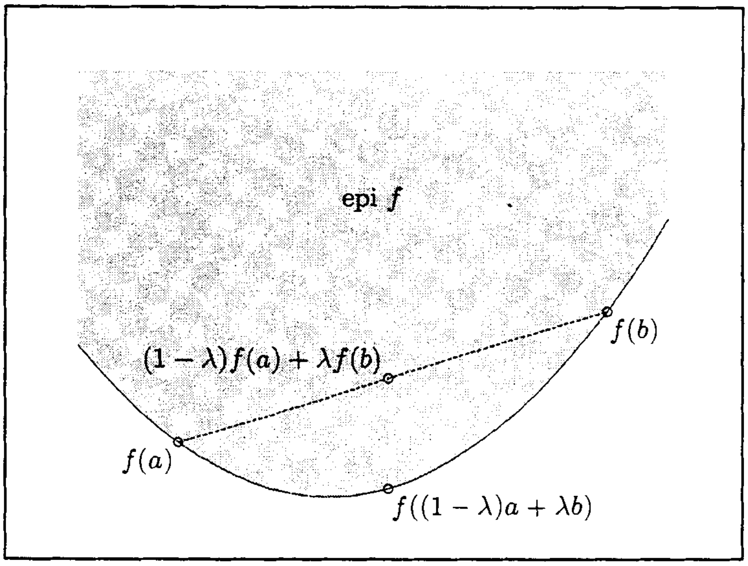


Figure 1.1: Interpolation characterization of convexity

Example 1.4 (Convexity of affine functions) Another example of a convex function is any affine function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, given by $f : x \mapsto \langle a, x \rangle + \alpha$. By linearity we have that $f((1-\lambda)x + \lambda y) = (1-\lambda)f(x) + \lambda f(y)$ and therefore f is convex by Theorem 1.3. □

This interpolation characterization of convexity is represented graphically in Figure 1.1, and an example of its utility is demonstrated in Example 1.4. Note that this characterization also brings rise to a stronger notion of convexity. A function is called *strictly convex* if the relation of Theorem 1.3 holds with strict inequality.

The definition of a convex function implies that the function is defined over a domain S which itself must be a convex set. To simplify the issue somewhat we may extend all functions to be defined over the whole space $\mathbf{E}$ by mapping them to the value $+\infty$ where they are not otherwise defined. This preserves the original structure of the function and allows us to exclude the explicit domain of the function from our definitions of convexity. This also allows us to recast problems like

$$\inf \{f(x) : x \in S\}$$

to a simpler representation of $\inf \{f(x) + \delta_S\}$.

Having extended functions to be defined over the whole space $\mathbf{E}$, we may sometimes wish to recapture the original domain of the function. We do so by redefining the *domain* of a function $f : \mathbf{E} \to \overline{\mathbb{R}}$ as the set

$$\operatorname{dom} f = \{x \in \mathbf{E} : f(x) < +\infty\}.$$

We say a function is *proper* if its domain is nonempty.

Convex functions naturally gives rise to other convex sets in various ways. One of the most important of these introduces the concept of level sets.

Theorem 1.5 (Convex level sets) *([16], Theorem 4.6, page 28) For any convex function $f : \mathbf{E} \to \overline{\mathbb{R}}$ and any $\alpha \in \mathbb{R}$, the level sets $\{x : f(x) < \alpha\}$, $\{x : f(x) \leq \alpha\}$, $\{x : f(x) > \alpha\}$ and $\{x : f(x) \geq \alpha\}$ are convex.*

Proof: The proof of this follows from Theorems 1.1 and 1.2 by observing that the level sets can by created by the intersection of the epigraph and the appropriate open or closed half-space, projected down to $\mathbf{E}$ from $\mathbf{E} \times \overline{\mathbb{R}}$. ■

1.3 Closures of Convex Functions

Many topological properties are implied directly by convexity. However, most of these results are made more accessible by introducing a little extra structure to the problem. A function $f : \mathbf{E} \to \overline{\mathbb{R}}$ is called *lower semi-continuous* on a set $S \subset \mathbf{E}$ at a point x if

$$f(x) \leq \lim_{i \to \infty} f(x^i)$$

for every sequence $x^1, x^2, \ldots$, in S such that $\lim x^i = x$, and the limit $\lim f(x^i)$, exists. This condition may alternatively be expressed as

$$f(x) \leq \liminf_{y \to x} f(y) = \liminf_{\epsilon \downarrow 0} \{f(y) : \|y - x\| \leq \epsilon\}.$$

Reversing the inequality leads to an equivalent definition for *upper semi-continuity.* Note that when f is finite on a neighborhood of x, the combination of both lower and upper semi-continuity at x implies continuity at x. The utility of lower semi-continuity is apparent from the theory of Fenchel conjugates (Section 1.6), and the following result.

Theorem 1.6 ([16], Theorem 7.1, page 51) *Consider a function $f : \mathbf{E} \to \overline{\mathbb{R}}$. Then the following conditions are equivalent:*

(a) f is lower semi-continuous on $\mathbf{E}$;

(b) $\{x : f(x) \leq \alpha\}$ is closed for every $\alpha \in \mathbb{R}$; and,

(c) the epigraph of f is a closed set.

Proof: Lower semi-continuity can be readily reexpressed as the condition that $u \geq f(x)$ whenever $u = \lim u^i$ and $x = \lim x^i$ for sequences $u^1, u^2, \ldots$, and $x^1, x^2, \ldots$, such that $u^i \geq f(x^i)$ for every i. Thus, any sequence of points $(x^1, u^1), (x^2, u^2), \ldots$, in the epigraph must have its limit in the epigraph, and we see that condition (a) is actually equivalent to condition (c). By taking $\alpha = u = u^1 = u^2 = \cdots$ we see that for any convergent sequence $x^1, x^2, \ldots$ such that $\alpha \geq f(x^i)$ it follows that $\alpha \geq f(x)$. In this manner, (a) implies (b). Now suppose that (b) holds, and we have sequences x^i converging to x and $f(x^i)$ converging to u. For every real $\alpha > u$, $f(x^i)$ must ultimately (for large enough i) be less than α, and thus

$$x \in \text{cl}\ \{y : f(y) \leq \alpha\} = \{y : f(y) \leq \alpha\}.$$

Hence we see that $f(x) \leq u$, and we see that (b) implies (a). ■

Given any function $f : \mathbf{E} \to \overline{\mathbb{R}}$, we define the *closure*, denoted cl f, as the function whose epigraph is itself the closure of epi f. A function is therefore said to be *closed* if cl $f = f$. Note that as implied by Theorem 1.6, for a proper convex function being closed is equivalent to being lower semi-continuous.

1.4 Continuity of Convex Functions

One of the most surprising results about convex functions is that the global geometric property of convexity can yield a local analytic property such as continuity. This result is explored in greater detail in the following theorems.

Lemma 1.7 (Bounded neighborhood) *([4], Lemma 4.1.2, page 66) Let Δ be the **simplex** $\{x \in \mathbb{R}^n_+ : \sum x_i \leq 1\}$. If a function $g : \Delta \to \mathbb{R}$ is convex, then it is bounded above on Δ.*

Proof: Let $\{e_1, \dots, e_n\}$ be the standard basis on $\mathbb{R}^n$. For any $x \in S$ it follows that $x = \sum_1^n x_i e_i$. By the convexity of g we see

$$\begin{aligned} g(x) &= g\left(\sum x_i e_i + (1 - \sum x_i)0\right) \\ &\leq \sum x_i g(e_i) + (1 - \sum x_i) g(0) \\ &\leq \max\{g(e_1), \dots, g(e_n), g(0)\}. \end{aligned}$$

Thus, g is bounded on Δ. ∎

Lemma 1.8 (Interior of epigraph) *Let x be a point in int dom f for a convex function f. Consider any point (x, ν) such that $\nu > f(x)$. Then $(x, \nu) \in$ int epi f.*

Proof: We present a geometric argument. Since $x \in$ int dom f there exists a scaled and translated simplex N such that $N \subset$ int dom f. By Lemma 1.7 it follows that f is bounded on N, and thus there exists a finite μ such that $f(y) \leq \mu$ for $y \in N$. By the definition of the epigraph, it follows that $C = \{(y, \mu) : y \in N\} \in$ epi f. Additionally, by convexity, it follows that the line segment from $(x, f(x))$ to any $y \in C$ is also in epi f, thus the vertical cone rooted at $(x, f(x))$ and extended to C is entirely within epi f. Similarly, the area extended above C is also entirely contained within epi f. Since (x, ν) lies along the central axis of this structure, we can always find a neighborhood around it contained completely within it, and therefore completely within int epi f. ∎

Note that the above Lemma actually holds in both directions, and any interior point of epi f can be used to find an interior point of int dom f. This stronger result can be found in Luenberger [13].

Theorem 1.9 (Continuity of convex functions) *Let $f : \mathbf{E} \to \overline{\mathbb{R}}$ be a convex function. Then f is continuous on int dom f.*

Proof: If f is improper, then f is identically ∞, and trivially continuous. Thus we may assume that f is proper and therefore finite on its non-empty int dom f.

In a proof parallel to that of Theorem 1.5 we can show that the upper level sets of cl epi f are all closed and (by the same logic as Theorem 1.6) that equivalently cl f is upper semi-continuous on int dom f. The combination of upper semi-continuity and lower semi-continuity from Theorem 1.6 shows that cl f is in fact continuous on dom cl f. It remains only to show that $f =$ cl f on int dom f.

Consider $x \in$ int dom f, and suppose cl $f(x) \neq f(x)$. Without loss of generality, shift our coordinates such that x is at the origin. Since epi $f \subset$ cl epi f, then by the definition of the epigraph this means that cl $f(0) < f(0)$. Let ν be a value arbitrarily close to but less than $f(0)$, such that $(0, \nu)$ is in cl epi f but not in epi f. Since $(0, \text{cl } f(x)) \in$ cl epi f we can construct a sequence $a^i = (x^i, \mu^i)$ such that $\lim a^i = (x, \text{cl } f)$, $\lim x^i = 0$, $\lim \mu^i =$ cl $f(0)$ and $a^i \in$ epi f. Consider the sequence of points $b^i = (-x^i, 2\nu - \mu^i)$. The sequence b^i approaches the point $b = (0, \lambda)$ where $\lambda > f(x)$. By Lemma 1.8 it follows that $b \in$ int epi f, thus for large enough i the sequence b^i is contained completely within epi f. Since $(a^i + b^i)/2 = (0, \nu)$, then by convexity $(0, \nu)$ is in epi f, a contradiction. Hence it must be that cl $f(x) = f(x)$ for all $x \in$ int dom f. Thus f is continuous on int dom f. ■

As shown in the above theorem, convexity of a function $f : \mathbf{E} \to \overline{\mathbb{R}}$ implies the continuity, and hence the lower semi-continuity, on the interior of the effective domain of f. Thus, in order for a function to be lower semi-continuous over the whole space $\mathbf{E}$ we need only concern ourselves with the definition of the function along the boundary of the domain. This suggests that lower semi-continuity is a natural form of normalization which makes convex functions more regular and easier to manipulate. It is therefore natural to restrict ourselves to the study to closed convex functions, incurring very little loss in generality. The functions then gain the three important properties outlined in Theorem 1.6.

Note that although convexity of a function f implies the continuity of f over the interior of its domain, it does not say anything about its differentiability. As an example, the one dimensional function $f : x \mapsto |x|$ is clearly convex, but it is not differentiable at the origin. However, given a function that is twice continuously differentiable on the interior of its domain, another characterization of convexity becomes useful. For simplicity we first examine the one dimensional case.

Theorem 1.10 (Second derivative characterization of convexity in 1D) ***([16], Theorem 4.4, page 26)*** *Consider $a < b \in \mathbb{R}$ and a function $f : (a, b) \to \overline{\mathbb{R}}$ that is twice continuously differentiable on (a, b). Then f is convex if and only if $f''(x)$ is non-negative on (a, b).*

Proof: Suppose f'' is non-negative on (a, b). Then it follows that f' is non-decreasing on (a, b). Taking $a < x < y < b$, $0 < \lambda < 1$ and $z = (1 - \lambda)x + \lambda y$, due to the non-decreasing derivative we have that

$$\begin{aligned} f(z) - f(x) &= \textstyle\int_x^z f'(t)dt \leq f(z)(z - x), \quad \text{and} \\ f(y) - f(z) &= \textstyle\int_z^y f'(t)dt \leq f(y)(y - z). \end{aligned}$$

Since $z - x = \lambda(y - x)$ and $y - z = (1 - \lambda)(y - x)$ we have

$$\begin{aligned} f(z) &\leq f(x) + \lambda f'(z)(y - x), \quad \text{and} \\ f(z) &\leq f(y) - (1 - \lambda)f'(y)(y - z). \end{aligned}$$

Multiplying the two inequalities by $(1 - \lambda)$ and λ respectively and adding them together yields

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

Thus, f is convex.

Suppose f'' is somewhere negative. Then by the continuity of f'', there exists some open sub-interval (a', b') on which f' is decreasing. By a similar argument to that just presented, it follows that f is strictly concave on this sub-interval, and therefore not convex. ■

A parallel result holds for continuously differentiable functions with non-decreasing first derivatives. This result will be explored a little further in the next section. Additionally, this one dimensional result can by extrapolated to n dimensions by taking one dimensional slices through a point in the direction of each basis vector of the higher space. If each slice through every point is convex, then the entire function is itself convex. We first introduce the notion of positive semidefinite and positive definite matrices.

Definition 1.11 (Positive Semidefinite) A matrix $M \in \mathbb{R}^{n \times n}$ is said to be *positive semidefinite* if $\langle x, Mx \rangle \geq 0$ for all $x \in \mathbb{R}^n$. Similarly, M is *positive definite* if $\langle x, Mx \rangle > 0$ for all non-zero $x \in \mathbb{R}^n$.

Theorem 1.12 (Hessian characterization of convexity) *([16], **Theorem 4.5, page 27**) Let $f : \mathbf{E} \to \overline{\mathbb{R}}$ be a twice continuously differentiable function defined on an open dom f. Then f is convex if and only if its Hessian matrix $H(x) = \nabla^2 f(x)$ is positive semidefinite everywhere in dom f.*

Proof: The convexity of f on $\mathbf{E}$ is equivalent to the convexity of the restriction of f to each line in $\mathbf{E}$. This is the same as the convexity of the function $g(t) = f(x + td)$ on $\mathbb{R}$ for each $x, d \in \mathbf{E}$. Vector calculus shows us that $g''(t) = \langle d, H(x + td)d \rangle$. Thus, by Theorem 1.10, $g(t)$ is convex for each $x, d \in \mathbf{E}$ if and only if $\langle d, H(y)d \rangle \geq 0$ for every $y, d \in \mathbf{E}$. ■

It is worth noting that the stronger condition of $H(x)$ being positive definite actually guarantees the strict convexity of f on a neighborhood of x. For more details, refer to [4].

1.5 Subgradients and the Subdifferential

The *directional derivative* of a function $f : \mathbf{E} \to \mathbb{R}$ at a point x in a direction $d \in \mathbf{E}$ is defined as

$$f'(x,d) = \lim_{t \downarrow 0} \frac{f(x+td) - f(x)}{t},$$

when this limit exists. If the directional limit $f'(x,d)$ is linear in d then there exists a (necessarily unique) vector $a \in \mathbf{E}$ such that $f'(x,d) = \langle a, d \rangle$. In this case we say that f is *(Gâteaux) differentiable* at x with *(Gâteaux) derivative* $\nabla f(x) = a$.

Standard calculus teaches us that a minimizer $\bar{x}$ of an everywhere differentiable function f is necessarily a critical point such that $\nabla f(\bar{x}) = 0$. However, many interesting convex functions are not everywhere differentiable which leads us to pursue different methods for representing derivative information. As an alternative to the derivative we instead consider the *subgradient*. A vector x^* is said to be a subgradient of a convex function $f : \mathbf{E} \to \overline{\mathbb{R}}$ at a point $x \in \mathbf{E}$ if

$$f(y) \geq f(x) + \langle x^*, y - x \rangle, \quad \forall y \in \mathbf{E}. \tag{1.13}$$

At points where the subgradient is defined, this *subgradient inequality* has a simple geometric interpretation: it says the affine function $f(x) + \langle x^*, y - x \rangle$ is a non-vertical supporting hyperplane to the convex set epi f at the point $(x, f(x))$. In the condition where f is differentiable at x it follows that the only such hyperplane is the one with slope defined by the gradient of f at x, in which case the only subgradient to f at x is $x^* = \nabla f(x)$. This geometric interpretation is demonstrated in Figure 1.2.

At points of non-differentiability it follows that there is more than one subgradient. This leads to the definition of the *subdifferential* of f at x as the set of all subgradients of f at x:

$$\partial f(x) := \{x^* : f(y) \geq f(x) + \langle x^*, y - x \rangle, \ \forall y \in \mathbf{E}\}.$$

The calculus-like relationship between subgradients and global minimizers is explored in the following theorem.

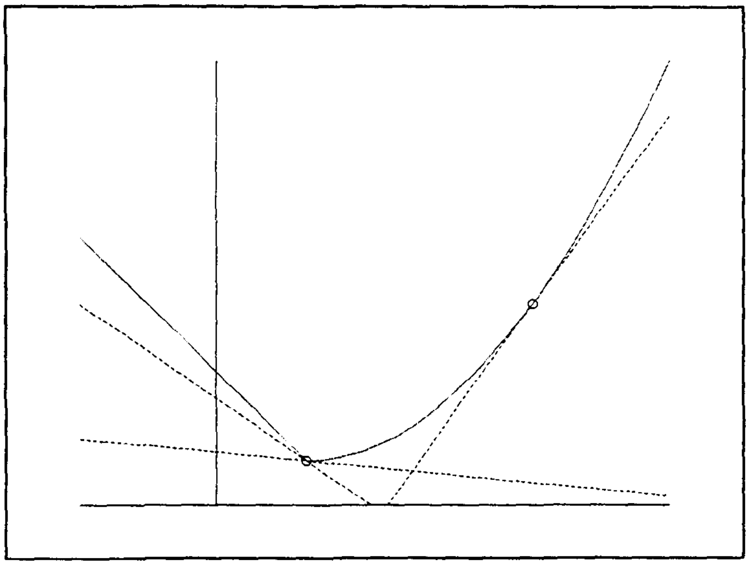


Figure 1.2: Some convex subgradients

Theorem 1.14 (Subgradients at global minimizers) *For any proper convex function $f : \mathbf{E} \to \overline{\mathbb{R}}$, the point x is a global minimizer of f if and only if the condition $0 \in \partial f(x)$ holds.*

Proof: This result follows from the definition of a subgradient in Equation 1.13. A global minimizer x must satisfy the relation $f(y) \geq f(x)$, for all $y \in E$. This is exactly the subgradient relationship for a point x with a vector $x^* = 0$. ■

Note the strong parallels between the theory of global minimizers for subdifferentials and of local minimizers for differentials. Furthermore, note that Theorem 1.14 reduces to the classical and familiar calculus result when f is everywhere differentiable over int dom f. The more subtle implication is that convex functions have a single local minimum; this is one of the properties that makes convex functions so attractive and tractable as optimization problems.

It is natural to begin by asking questions about the existence and general behaviour of directional derivatives on convex functions. Some key properties of these functions are presented in the following theorem.

Theorem 1.15 (Existence of directional derivatives) *([16], **Theorem 23.1, page 215**) Let f be a convex function and let x be a point in int dom f. For each d, the difference quotient in the definition of $f'(x,d)$ is a non-decreasing function of $t > 0$, so that $f'(x,d)$ exists. Moreover, $f'(x,\cdot)$ is convex, $f'(x,0) = 0$ and $-f'(x,-d) \leq f'(x,d)$, for all d.*

Proof: For simplicity let $h(y) = f(x+y) - f(x)$ so that the difference quotient may be compactly expressed as $t^{-1}h(td)$. The set epi h is the translate of epi f with $(x, f(x))$ moved to the origin, and is therefore also convex. On the other hand, we may also write $t^{-1}h(td) = (ht^{-1})(d)$, where by definition ht^{-1} is the convex function whose epigraph is t^{-1}epi h. Since epi h contains the origin, the latter set increases, if anything, as t^{-1} increases. In other words, for each d, the difference quotient $(ht^{-1})(d)$ can only possibly decrease as t decreases. Hence the limit in the directional derivative is bounded below and must exist.

Since $f'(x,\cdot)$ is defined as the limit of a sequence of convex functions, it too must be convex. Moreover, by the definition of the directional derivative, we see that $f'(x,0) = 0$. Finally, by the convexity of $f'(x,\cdot)$ one has

$$\frac{1}{2}f'(x,-d) + \frac{1}{2}f'(x,d) \geq f'\left(x, \frac{1}{2}(-d+d)\right) = f'(x,0) = 0,$$

and therefore $-f'(x,-d) \leq f'(x,d)$, for all d. ■

It is clear that there is an intimate relationship between directional derivatives and subgradients. This relationship is formalized in the following theorem, adapted from [16].

Theorem 1.16 (Directional derivatives and subgradients) *Consider a convex function $f : \mathbf{E} \to \overline{\mathbb{R}}$. Then x^* is a subgradient of f at $x \in$ int dom f if and only if*

$$f'(x,d) \geq \langle x^*, d\rangle, \quad \forall d.$$

Proof: Suppose that x^* is a subgradient of f at x. Setting $y = x + td$ we can rewrite the subgradient inequality (Equation 1.13) as

$$\frac{f(x+td) - f(x)}{t} \geq \langle x^*, d\rangle, \quad \forall t > 0, \forall d \in \mathbf{E}.$$

Since the difference quotient decreases to $f'(x,t)$ in the limit as t decreases to zero we are left with the desired inequality from the theorem.

Suppose the directional limit inequality holds. By the convexity of f and the non-decreasing nature of its directional derivatives from Theorem 1.15, we see that $f(y) \geq f(x) + f'(x, y - x)$. A direct substitution yields that $f(y) \geq f(x) + \langle x^*, y - x\rangle$, which is exactly the subgradient inequality. ■

In the one dimensional case of the above theorem the subgradients are the slopes x^* of the non-vertical lines in $\mathbb{R}^2$ which pass through $(x, f(x))$ without meeting int epi f. These form the closed interval of real numbers between $f'_-(x) = -f'(x, -1)$ and $f'_+(x) = f'(x, +1)$. We will revisit and formalize this result a little later. We first solidify the relationship between differentials and subgradients in the following theorem.

Theorem 1.17 (Differentiability of convex functions) *([16], Theorem 25.1, page 242) Consider the proper convex function $f : \mathbf{E} \to \overline{\mathbb{R}}$. Then the function f is Gâteaux differentiable at a point $x \in$ int dom f if and only if f has a unique subgradient x^* at x (in which case $\partial f(x) = \{x^*\} = \{\nabla f(x)\}$).*

Proof: Suppose that f is differentiable at x. Then from the definition of differentiability there exists a unique vector a such that $f'(x, d) = \langle a, d\rangle$. Substituting this into Theorem 1.16 yields the inequality

$$\langle a, d\rangle \geq \langle x^*, d\rangle, \quad \forall d \in \mathbf{E}.$$

The only way this can hold for all d is with equality when $x^* = a$, thus $a = \nabla f(x)$ is the only subgradient of f at x.

Suppose that f has a unique subgradient at x. Without loss of generality, we may consider the translated scaled function g such that $g(y) = f(x + y) - f(x) - \langle x^*, y\rangle$. This function will have the unique subgradient 0 at the origin, and we must show that

$$\lim_{y \to 0} \frac{g(y)}{\|y\|} = 0.$$

Suppose that there exists a direction d such that $g'(0, d) = \mu \neq 0$. Let $m = \mu d/\|d\|^2$, then $\langle m, d\rangle = \mu$. It follows that $g(td) \geq \langle m, td\rangle$. Similarly, by Theorem 1.15 we have that $g'(0, -d) \leq -\mu = \langle m, -d\rangle$ thus $g(-td) \geq \langle m, -td\rangle$. For any e perpendicular to d it follows that $\langle m, e\rangle = 0$, thus $g(te) \geq \langle m, te\rangle$. By the convexity of g it follows that for any y, $g(y) \geq \langle m, y\rangle$. However, this means that m is also a subgradient, a contradiction. It must therefore be that $g'(0, d) = 0$ for all d.

Let $h_\lambda(u) = g(\lambda u)/\lambda$. Let $\{a_1, \dots, a_n\}$ be any finite collection of points whose convex hull contains the ball B. Each $u \in B$ may be expressed as $u = \lambda_1 a_1 + \cdots + \lambda_n a_n$, and it follows that

$$\begin{aligned} 0 \leq h_\lambda(u) &\leq \sum_{i=1}^{n} \lambda_i h_\lambda(a_i) \\ &\leq \max\{h_\lambda(a_i) : i = 1, \dots, n\}. \end{aligned}$$

Since $h_\lambda(a_i)$ decreases to 0 for each i as $\lambda \downarrow 0$, it follows that $h_\lambda(u)$ does likewise. Hence, given any $\epsilon > 0$ there exists a $\delta > 0$ such that

$$g(\lambda u)/\lambda \leq \epsilon, \quad \forall \lambda \leq \delta, \quad \forall u \in B.$$

Since each vector y with $\|y\| \leq \delta$ may be written as λu for some $u \in B$, we have that $g(y)/\|y\| \leq \epsilon$. Hence, the limit of $g(y)/\|y\|$ is 0, and thus the zero vector is by definition the gradient of g at the origin. ■

Note that we are actually proving the stronger notion of *Fréchet differentiability* here. This is not completely surprising as on the interior of the domain of convex functions f defined over $\mathbb{R}^n$, f is locally Lipschitz and these two notions of differentiability become equivalent. For more details refer to Chapter 6 of [4].

As alluded to earlier, the situation is vastly simplified in one-dimension. If a function $f : \mathbb{R} \to \overline{\mathbb{R}}$ is proper and convex, by Theorem 1.15 the directional derivatives exist at every point in the interior. Theorem 1.16 gave us come clues as to how to completely formulate the subgradient of a one dimensional function, and we formalize that result in our next theorem.

Theorem 1.18 (Subdifferential in one dimension) *Consider a proper convex function $f : \mathbb{R} \to \overline{\mathbb{R}}$. For each point $x \in$ int dom f the subdifferential is given by the (potentially singleton) closed interval*

$$\partial f(x) = [f'_-(x), f'_+(x)].$$

Furthermore, the subdifferential is a singleton only at those points x where f is differentiable.

Proof: Consider any points x at which f is differentiable. At these points, $f'_-(x) = f'_+(x) = \nabla f(x)$ and the above set is a singleton equal to $\{\nabla f(x)\}$, which is the subdifferential of f at x by Theorem 1.17.

Consider now any points in x at which f is not differentiable. We must have that $f'_-(x) \neq f'_+(x)$, and by Theorem 1.15 we have specifically that $f'_-(x) < f'_+(x)$. Consider $x^* \in [f'_-(x), f'_+(x)]$. It follows that $f'_-(x) \leq x^* \leq f'_+(x)$, and therefore

$$\begin{aligned} f'(x,-1) &\geq -x^*, \quad \text{and} \\ f'(x,1) &\geq x^*. \end{aligned}$$

Thus, by Theorem 1.16 it follows that x^* is a subgradient of f at x. Additionally, inspection shows that there can be no other x^* that satisfy the system of two linear inequalities from Theorem 1.16, thus we may represent all of the subgradients of f at x as $\partial f(x) = [f'_-(x), f'_+(x)]$. ■

We finish this section with an example illustrating a practical application of Theorem 1.18.

Example 1.19 (Subgradient of absolute value function) Consider the function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = |x|$. This function is differentiable everywhere but at the origin, thus by Theorem 1.17, $\partial f(x) = \{f'(x)\}$, for all $x \neq 0$. The left derivative at the origin is easily calculated as $f'_-(x) = -1$, while the right derivative is calculated as $f'_+(x) = 1$. Using Theorem 1.18 the entire subdifferential is therefore given by

$$\partial f(x) = \begin{cases} \{-1\}, & x < 0 \\ [-1,1], & x = 0 \\ \{1\}, & x > 0. \end{cases}$$

□

1.6 The Fenchel Conjugate

As characterized in Equation 1.13 we may view a convex function as being minorized at each finite point $f(x)$ by at least one unique non-vertical hyperplane. This leads to a natural alternative representation of a convex function as being defined by the envelope of its tangent hyperplanes. Equivalently, we can consider the epigraph of the function as being defined by the closed-halfspaces which contain it. This concept is captured in the following result from Rockafeller [16].

Theorem 1.20 (Envelope representation of convex functions) *(**[16], Theorem 12.1, page 102**) A proper closed convex function f is the pointwise supremum of the collection of all affine functions h such that $h \leq f$.*

Proof: Since epi f is a closed convex set it may be visualized as the intersection of all half-spaces containing it. The half-spaces can not all be vertical since that would imply that epi f was a union of vertical lines, contrary to properness. There is a one-to-one correspondence between each non-vertical half-space and a minorizing affine function describing the half-space. The non-vertical half-spaces are the epigraphs of the corresponding affine functions. To prove the theorem we must show that the vertical half-spaces (who have no affine function counterpart) are redundant in defining f. In other words, given any vertical half-space V containing epi f and a point v outside of V, find a minorizing affine function h that excludes the point v. Let $V = \{(x,u) : 0 \geq \langle x, b_1\rangle - \beta_1 = h_1(x)\}$ and let $v = (x_0, u_0)$. By the properness of f we know there exists at least one minorizing affine function h_2 such that $h_2 \leq f$. For every $x \in \text{dom } f$ we have $h_1(x) \leq 0$ and $h_2(x) \leq f(x)$, and thus

$$\lambda h_1(x) + h_2(x) \leq f(x), \quad \forall \lambda \geq 0.$$

The same inequality holds when $x \notin \text{dom } f$ because then $f(x) = \infty$. Thus, for any $\lambda > 0$ we may define h as

$$h(x) = \lambda h_1(x) + h_2(x) = \langle x, \lambda b_1 + b_2\rangle - (\lambda\beta_1 + \beta_2)$$

and have an affine function h such that $h \leq f$. Since $h_1(x_0) > 0$, choosing λ sufficiently large will ensure that $u_0 < h(x_0)$ as desired. ■

Corollary 1.21 (Existence of minorizing hyperplanes) *Given a proper convex function $f : \mathbf{E} \to \overline{\mathbb{R}}$ there exists some $b \in \mathbf{E}$ and $\beta \in \overline{\mathbb{R}}$ such that $f(x) \geq \langle x, b\rangle - \beta$ for every x.*

According to Theorem 1.20 there is a dual way of describing any closed convex function f on $\mathbf{E}$: we can describe the set F^* consisting of all pairs (x^*, μ^*) in $\mathbf{E} \times \mathbb{R}$ such that the affine function $h(x) = \langle x, x^*\rangle - \mu^*$ is majorized by f. It follows that $h(x) \leq f(x)$ for all x if and only if

$$\mu^* \geq \sup\{\langle x, x^*\rangle - f(x)\}.$$

Thus F^* is the epigraph of the function f^* defined by

$$f^*(x^*) = \sup_x \{\langle x, x^* \rangle - f(x)\}. \tag{1.22}$$

This f^* is called the *Fenchel conjugate* of f (sometimes referred to as the *Fenchel-Legendre transform*). This function can be viewed as the pointwise supremum of the collection of affine functions $g(x^*) = \langle x, x^* \rangle - \mu$ such that (x, μ) belongs to $F = \text{epi } f$. As such, f^* is itself another closed convex function. In a parallel relationship, we see that f may itself be defined as the pointwise supremum of the affine functions $h(x) = \langle x, x^* \rangle - \mu^*$ such that $(x^*, \mu^*) \in F^* = \text{epi } f^*$, and therefore

$$f(x) = \sup_{x^*} \{\langle x, x^* \rangle - f^*(x^*)\} = f^{**}(x).$$

Inspection shows that the conjugacy operation of Equation 1.22 is order-reversing; that is, for functions $f, g : E \to \overline{\mathbb{R}}$ the inequality $f \geq g$ implies that $f^* \leq g^*$.

Example 1.23 (Absolute value function) Consider the function $f : \mathbb{R} \mapsto \mathbb{R}$ defined by $f(x) = |x|$ for all $x \in \mathbb{R}$. By definition the conjugate is given by

$$f^*(y) = g(y) = \sup_x \{xy - |x|\}.$$

Splitting the function at the origin yields the following

$$\begin{aligned} g(y) &= \max\left\{\sup_{x \leq 0}\{x(y+1)\}, \sup_{x>0}\{x(y-1)\}\right\} \\ &= \max\left\{\begin{cases} +\infty, & y < -1 \\ 0, & y \geq -1 \end{cases}, \begin{cases} 0, & y \leq 1 \\ +\infty, & y > 1 \end{cases}\right\} \\ &= \begin{cases} 0, & y \in [-1, 1] \\ +\infty, & \text{otherwise.} \end{cases} \end{aligned}$$

□

Finding the conjugate at a point y can be visualized as finding the point $\bar{x}$ at which the hyperplane of slope y is furthest above the convex function f. When this supremum is attained and unique, we may shift the hyperplane of slope y down by the value $f^*(y)$ and visualize a minorizing hyperplane $h(x) = \langle x, y \rangle - f^*(y)$ touching the original function $f(x)$ at $\bar{x}$. This allows us to take the alternative view that the conjugate value of a function f at

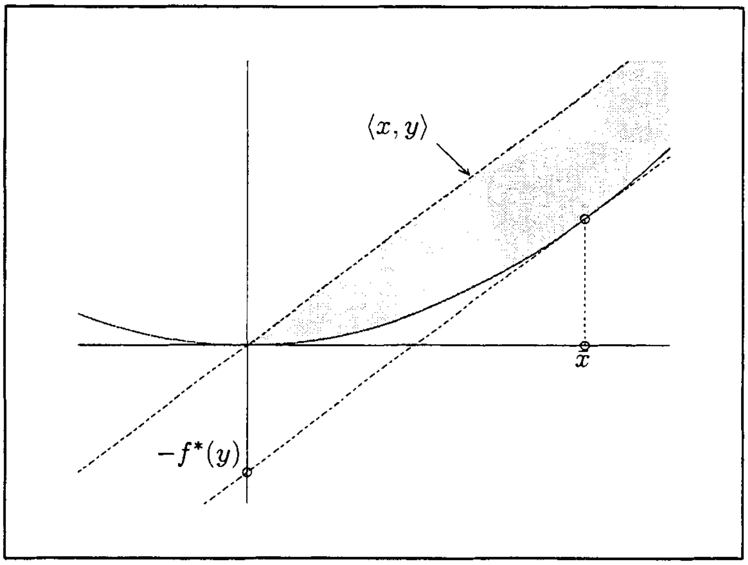


Figure 1.3: Vertical intercept interpretation of conjugate

a point y is equal to the negative of the value at the origin of the maximum hyperplane of slope y that minorizes f (in other words, which is a subgradient of f at the point $\bar{x}$). This interpretation of the conjugate is shown graphically in Figure 1.3.

An immediate consequence of the definition of the Fenchel conjugate is the well-known *Fenchel-Young inequality* .

Theorem 1.24 (Fenchel-Young inequality) *Given a function* $f : \mathbf{E} \to \overline{\mathbb{R}}$ *and* $x \in$ *dom* f*, the following inequality holds for all* $x^* \in \mathbf{E}$

$$f(x) + f^*(x^*) \geq \langle x, x^* \rangle.$$

Moreover, the preceding holds with equality if and only if

$$x^* \in \partial f(x).$$

Proof: The inequality is immediate from the definition of the Fenchel conjugate in Equation 1.22:

$$\begin{aligned} f^*(x^*) &= \sup_x \{\langle x, x^* \rangle - f(x)\} \\ &\geq \langle x, x^* \rangle - f(x). \end{aligned}$$

By the definition of the subdifferential (Equation 1.13), $x^* \in \partial f(x)$ holds if and only if

$$f(y) \geq f(x) + \langle x^*, y - x\rangle$$

or, equivalently

$$\langle x^*, y\rangle - f(y) + f(x) \leq \langle x^*, x\rangle$$

for all $y \in \mathbf{E}$. Taking the supremum over all y this is equivalent to

$$f^*(x^*) + f(x) \leq \langle x^*, x\rangle$$

which proves the result. ■

As earlier discussed, all closed convex functions f equal their biconjugates f^{**}. These functions naturally occur as pairs with proper closed convex functions having conjugates which are themselves proper and closed convex functions. We consider now the special case of self-conjugate functions.

Theorem 1.25 (Self-conjugate functions) *Consider a proper closed convex function $f : \mathbf{E} \to \overline{\mathbb{R}}$ such that $f^* = f$. Then $f(x) = \frac{1}{2}\langle x, x\rangle$.*

Proof: Consider the function $x \mapsto \frac{1}{2}\langle x, x\rangle$. The Fenchel conjugate of this function is given by $\sup_x\{\langle x, y\rangle - \frac{1}{2}\langle x, x\rangle\} = \sup_x\{\sum(x_i y_i - \frac{1}{2}x_i^2)\} = \sum \sup_{x_i}\{x_i y_i - \frac{1}{2}x_i^2\}$. Taking the derivative of the inner function yields $y_i - x_i$, thus the maximum occurs at $x_i = y_i$. Substituting this back into the equation yields the conjugate $\frac{1}{2}\langle x, x\rangle$. Thus, we see that $\frac{1}{2}\langle x, x\rangle$ is self-conjugate.

Suppose we have a function f such that $f = f^*$. Then by Theorem 1.24 it follows that $f(x) \geq \frac{1}{2}\langle x, x\rangle$. Since conjugation is an order-reversing operation, it also follows that $f^*(x) \leq (\frac{1}{2}\langle x, x, \rangle)^*$, or equivalently $f(x) \leq \frac{1}{2}\langle x, x\rangle$. Thus it must be that $f(x) = \frac{1}{2}\langle x, x\rangle$. ■

By the above theorem, it is now evident that there is only one function that is self-conjugate, and that all other conjugate pairs must therefore consist of two distinct functions. Refer to Table 1.1 for a brief list of some convex functions and their Fenchel conjugates.

$f(x) = g^*(x)$	dom f	$g(y) = f^*(y)$	dom g
0	$\mathbb{R}$	0	$\{0\}$
$bx + c$	$\mathbb{R}$	$-c$	$\{b\}$
x	$\mathbb{R}_+$	0	$[0, 1]$
$\|x\|$	$\mathbb{R}$	0	$[-1, 1]$
$\|x\|^p/p,\ p > 1$	$\mathbb{R}$	$\|y\|^q/q \quad (\frac{1}{p} + \frac{1}{q} = 1)$	$\mathbb{R}$
e^x	$\mathbb{R}$	$\begin{cases} 0, & y = 0 \\ y \ln y - y, & y > 0 \end{cases}$	$\mathbb{R}_+$
$-\log x$	$\mathbb{R}_{++}$	$-1 - \log -y$	$-\mathbb{R}_{++}$

Table 1.1: Some conjugate pairs of one dimensional convex functions

1.6.1 Concave Functions

All of the theory developed up until this point can be analogously applied to concave functions, with minor modifications. It should be noted that concave functions are not best handled by multiplying by -1 and using the appropriate convex machinery, but rather through a completely parallel theory. We cover the salient points here.

Consider a concave function g defined over a convex subset S of the space $\mathbf{E}$. As with convex functions, we can easily extend this function to the whole space by defining it to take the value of $-\infty$ outside of S. Similarly, we may define the *hypograph* of f to be the set

$$\text{hyp } g = \{(x, \lambda) \in \mathbf{E} \times \mathbb{R} : \lambda \leq g(x)\}.$$

The notion of a subgradient may be replaced with a similar notion of a supergradient, and the Fenchel conjugate for concave functions may be appropriately defined as

$$g_*(x_*) = \inf_x \{\langle x_*, x \rangle - g(x)\}.$$

The geometric interpretation of the concave conjugate is similar to that for convex conjugates. The hyperplane $\langle x_*, x \rangle - r = g_*(x_*)$ majorizes the set hyp g, and $-g_*(x_*)$ is its vertical intercept. The situation is summarized in Figure 1.4. Furthermore, it can be seen that the concave conjugate is related to the convex conjugate in the following manner:

$$g_*(x) = -(-g)^*(-x).$$

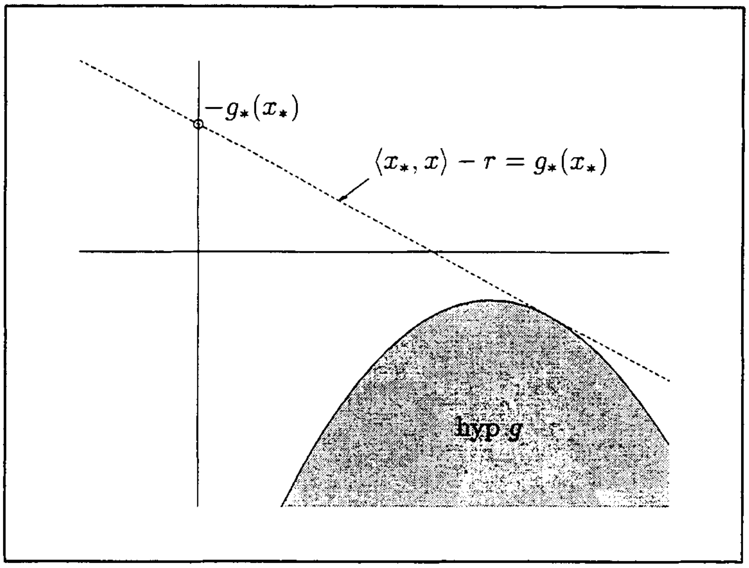


Figure 1.4: Conjugate relationship for concave functions

All of the results proved earlier have concave counterparts of the same form, usually involving only a change in the direction of inequality. We will use these results without explicit proof.

1.7 Fenchel Duality

The theory of Fenchel duality exists in various forms, but we will present here the traditional symmetric problem as described in [13, 16]. Newer works such as [4, 5] describe related but slightly more general duality results involving systems with linear constraints.

Suppose we seek to minimize the difference between a convex function and a concave function. Given a convex function f and a concave function g this amounts to solving

$$\inf_x \{f(x) - g(x)\}.$$

In a typical convex optimization problem g is uniformly zero (indeed, $f(x) - g(x)$ is itself a convex function), but this generalized form of the problem is conceptually useful. The problem can be interpreted as finding the minimum vertical distance between the sets epi f and hyp g. Imagine vertically displacing epi f until it just touches hyp g. At the point

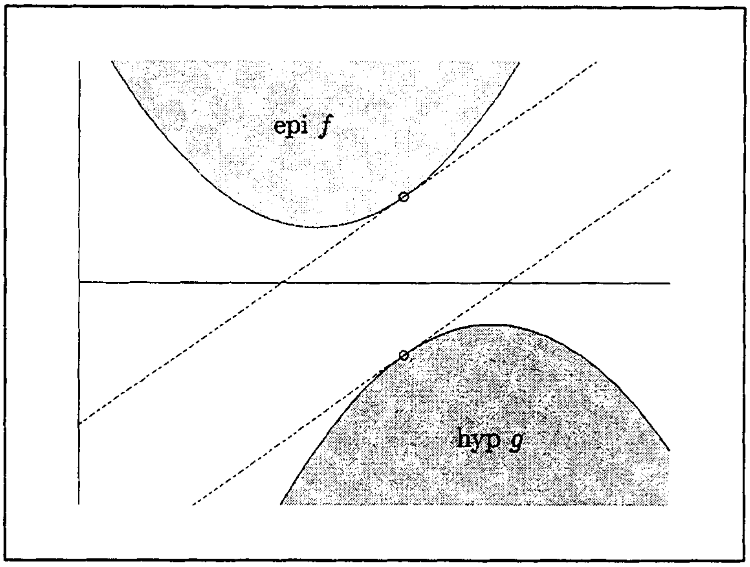


Figure 1.5: Fenchel duality

of contact these sets may be separated by a (not necessarily unique) hyperplane. Thus, geometric intuition tells us that we can consider the minimum vertical distance between f and g as being equivalent to the maximum vertical distance between parallel supporting hyperplanes that separate f and g.

The conjugate plays a natural role in expressing this dual relationship algebraically. Since $-f^*(y)$ is the vertical intercept of the support hyperplane of slope y minorizing epi f and $-g_*(y)$ is the vertical intercept of the support hyperplane of slope y majorizing hyp g, it follows that $g_*(y) - f^*(y)$ is the vertical seperation between the two parallel hyperplanes. This duality is illustrated in Figure 1.5 and detailed in the following theorem.

Theorem 1.26 (Fenchel duality theorem) ***([13], Section 7.12, Theorem 1, page 201)*** *Assume that f and g are, respectively, convex and concave functions defined on* $\mathbf{E}$. *Assume that $C =$ int dom $f \cap$dom g is non-empty. Suppose further that the the minimization*

$$\mu = \inf_x \{f(x) - g(x)\}$$

is finite. Then it follows that

$$\sup_y \{g_*(y) - f^*(y)\}$$

will attain a finite maximum of μ achieved by some $\bar{y} \in D = dom\ g_* \cap dom\ f^*$.

Additionally, if the primal infimum is attained by a point $\bar{x} \in C$, then

$$\sup_x \{\langle x, \bar{y}\rangle - f(x)\} = \langle \bar{x}, \bar{y}\rangle - f(\bar{x})$$

and

$$\inf_x \{\langle x, \bar{y}\rangle - g(x)\} = \langle \bar{x}, \bar{y}\rangle - g(\bar{x}).$$

Proof: By definition, for all $x \in C$ and $y \in D$ we see that

$$\begin{aligned} f^*(y) &\geq \langle y, x\rangle - f(x), \text{ and} \\ g_*(y) &\leq \langle y, x\rangle - g(x). \end{aligned}$$

Therefore

$$f(x) - g(x) \geq g_*(y) - f^*(y)$$

and hence

$$\inf_x \{f(x) - g(x)\} \geq \sup_y \{g_*(y) - f^*(y)\}.$$

The equality in the theorem can be proved if a $\bar{y} \in D$ can be found for which $\inf_x \{f(x) - g(x)\} = g_*(\bar{y}) - f^*(\bar{y})$.

By the definition of μ the convex sets epi $\{f - \mu\}$ and hyp g are arbitrarily close, but with disjoint interiors. Since these sets have non-empty interior there exists a non-vertical hyperplane in $\mathbf{E} \times \mathbb{R}$ separating them which may be represented as $\{(x, r) : \langle \bar{y}, x\rangle - r = c\}$ for some $\bar{y} \in D$ and $c \in \mathbb{R}$ (a vertical hyperplane would imply int dom $f \cap$ int dom $g = \emptyset$, a contradiction). Since hyp g lies below this hyperplane but arbitrarily close to it, we have

$$c = \inf_x \{\langle \bar{y}, x\rangle - g(x)\} = g_*(\bar{y}).$$

By a similar argument, it is seen that

$$c = \inf_x \{\langle \bar{y}, x\rangle - f(x) + \mu\} = f^*(\bar{y}) + \mu,$$

and therefore $\mu = g_*(\bar{y}) - f^*(\bar{y})$.

If the infimum μ is attained by some $\bar{x} \in C$ then the set epi $\{f - \mu\}$ and hyp g have the point $(g(\bar{x}), \bar{x})$ in common. This point lies in the separating hyperplane and gives the two final equalities. ■

1.7.1 Examples of Fenchel Duality

Several other duality results can be seen to be implied by Fenchel duality. One example of this is the well known linear programming duality theorem, stated below. For a proof of this theorem and many further results regarding linear programming, refer to [17].

Theorem 1.27 (Linear programming duality) *Consider a primal linear program*

$$\min_x\{\langle c, x\rangle : x \geq 0, Ax = b\}$$

and its dual

$$\max_y\{\langle b, y\rangle : A^*y \leq c\}.$$

Exactly one of the following holds:

- *the primal attains its optimal solution, in which case so must the dual, and their objective values are equal;*
- *the primal is infeasible, in which case the dual is either unfeasible or unbounded; or,*
- *the primal is unbounded, in which case the dual is infeasible.*

Example 1.28 (Linear programming duality) Consider the following primal linear program:

$$\min_x\{\langle c, x\rangle : x \geq 0, Ax = b\},$$

where $c \in \mathbb{R}^n$, $b \in R^m$, and $A \in R^{m\times n}$. This problem is easily recast into the framework of Fenchel duality by first defining

$$f(x) = \begin{cases} \langle c, x\rangle, & x \geq 0 \\ \infty, & \text{otherwise.} \end{cases}$$

This f is convex on $\mathbb{R}^n$. Secondly, we define a concave indicator function g as

$$g(x) = \begin{cases} 0, & Ax = b \\ -\infty, & \text{otherwise.} \end{cases}$$

We can easily see that f and g yield a Fenchel primal problem that is equivalent to the Linear Programming primal.

Straight-forward computation of conjugates yields

$$f^*(x^*) = \begin{cases} 0, & x^* \leq c \\ \infty, & \text{otherwise} \end{cases} \quad \text{and} \quad g_*(x_*) = \inf_x\{\langle x, x_*\rangle : Ax = b\}$$

and the dual Fenchel problem

$$\sup_z\{\inf_x\{\langle x, z\rangle : Ax = b\} : z \leq c\}.$$

Making the substitution $z = A^*y$ for $y \in \mathbb{R}^m$ yields

$$\sup_y\{\inf_x\{\langle x, A^*y\rangle : Ax = b\} : A^*y \leq c\}.$$

Since $\langle x, A^*y\rangle = \langle Ax, y\rangle$ then this is further simplified to

$$\sup_y\{\langle b, y\rangle : A^*y \leq c\},$$

which is precisely the linear programming dual.

Fenchel duality yields the linear programming primal/dual relationship, but it is not strong enough to guarantee that there is not any duality gap when the primal program attains its optimum. In order to fully recover linear programming duality we have to appeal to results based on the polyhedrality of the primal domain $\{x : Ax = b\}$. For further details on this, refer to Chapter 5 of [4]. □

In a similar manner the classical Min-Max theorem of game theory may be fully recovered as an example of Fenchel duality. The following result is presented in [13].

Theorem 1.29 (Min-Max) *Let A and B be non-empty compact convex subsets of* $\mathbf{E}$. *Then*

$$\min_{x\in A}\max_{y\in B}\langle x, y\rangle = \max_{y\in B}\min_{x\in A}\langle x, y\rangle.$$

Proof: Define the function f on $\mathbf{E}$ as

$$f(x) = \max_{y\in B}\langle x, y\rangle.$$

This maximum exists and is attained for every $x \in X$ since B is compact. The function is easily shown to be convex and continuous on $\mathbf{E}$. Let $g = -\delta_A$. The Fenchel primal problem arising from these functions is therefore

$$\min_{x\in A}\{f(x)\},$$

which exists by the compactness of A and the convexity of f. We now apply the Fenchel duality theorem, yielding

$$g_*(y) = \min_{x\in A}\langle x, y\rangle$$

by the definition of the concave conjugate. Consider δ_B. The convex conjugate of this functional is given by

$$\begin{aligned}(\delta_B)^*(y) &= \max_x\{\langle x, y\rangle - \delta_B(x)\} \\ &= \max_{x\in B}\langle x, y\rangle \\ &= f(y).\end{aligned}$$

We see that δ_B and f are a conjugate pair, thus $f^* = \delta_B$. The dual then becomes

$$\max_{y\in B} g_*(y) = \max_{y\in B}\min_{x\in A}\langle x, y\rangle.$$

The final result comes directly from the equivalence of the two expressions under Fenchel duality. ■

Notice that in this example, the compactness of the solution space allowed us to guarantee that solutions exist and objective values are attained. Because of the potentially unbounded or infeasible nature of linear programs, this was not possible in the previous example, hence the weaker result.

Chapter 2

Convex Analysis in One Dimension

In this chapter we explore the problem of calculating Fenchel conjugates symbolically for functions defined on the real line. We begin with an overview of the work presented in [2, 3], and present extensions to that work that enable it to operate on a broader class of functions.

2.1 A Good Class of Functions

Computer algebra systems are naturally suited to working with functions defined over the real numbers that are finite in representation. It is useful to characterize what we mean by having a finite representation, and to formalize the space of admissible functions.

Let $\mathcal{F}$ be the class of all functions f satisfying the following conditions:

(i) f is a function from $\mathbb{R}$ to $\overline{\mathbb{R}}$;

(ii) f is a closed convex function;

(iii) f is continuous on its effective domain; and,

(iv) there are finitely many points x_i such that $x_0 = -\infty < x_1 < \cdots < x_{n-1} < x_n = \infty$ and f restricted to each open interval is one of the following:

  (a) identically equal to ∞; or,

  (b) differentiable.

The class of functions $\mathcal{F}$ encompasses all closed convex functions that are naturally representable (piecewise with finitely many breaks) in a computer algebra system. In this manner, it is very well suited to our purpose. Additionally, it is easily seen that $\mathcal{F}$ is closed under positive scalar multiplication, and addition. As will be shown later, for a given $f \in \mathcal{F}$, f^* can have at most finitely many points of non-differentiability, thus $\mathcal{F}$ is also closed under the operation of conjugation.

2.2 Subdifferentiation

Subdifferentiation of functions in the class $\mathcal{F}$ is not very different from calculating standard univariate derivatives. In the case where f is a proper convex one dimensional function, we may calculate the subdifferential directly as outlined in Theorem 1.18, with the subdifferential being undefined outside of dom f. The remaining two improper cases are easily handled as exceptions to the general rule.

The algorithm begins by calculating the derivative f_i' along each open interval (x_i, x_{i+1}) in int dom f, which yields the subdifferential by Theorem 1.18. Next, the left and right derivatives are calculated at each point $x_i \in$ int dom f, with the subdifferential at these points given by the (possibly singleton) closed set

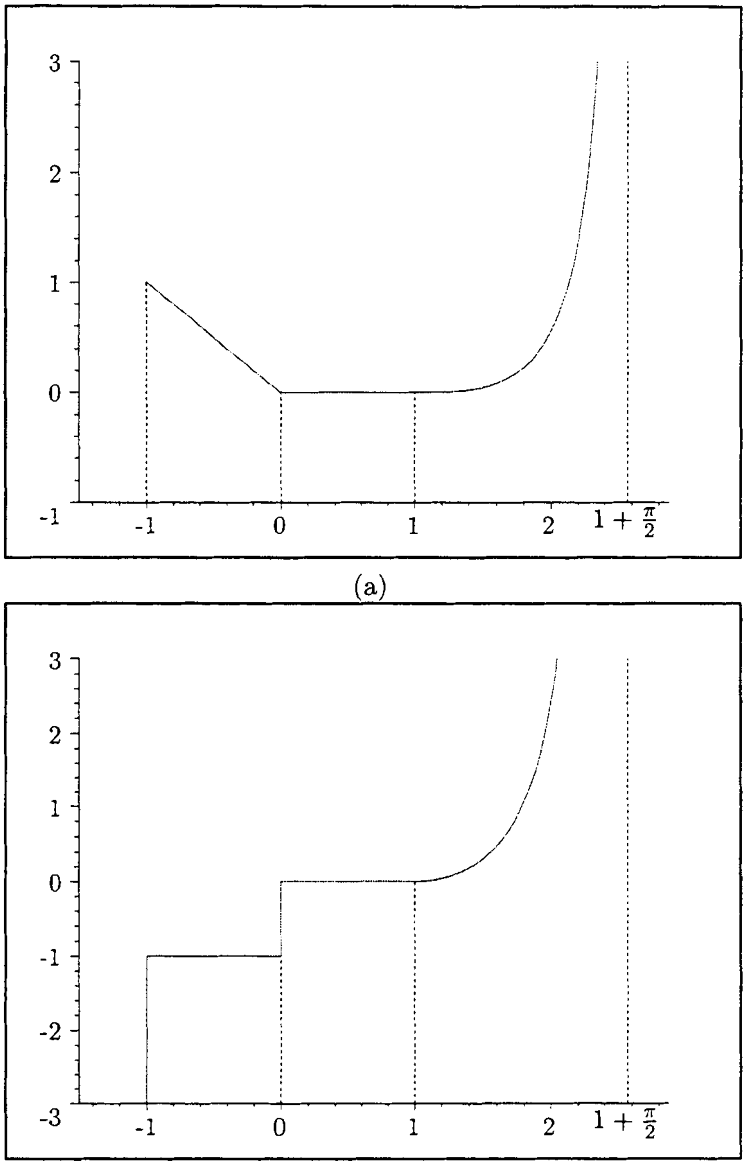
$$\left[\lim_{x \uparrow x_i} f_i'(x),\ \lim_{x \downarrow x_i} f_{i+1}'(x)\right].$$

For x_i not in dom f, the subdifferential is defined to be empty; the remaining cases involving points in bd dom f, which are not covered under Theorem 1.18, are best illustrated in an example.

Example 2.1 Consider the following function, illustrated in Figure 2.1(a):

$$f(x) = \begin{cases} \infty, & -\infty < x < -1 \\ -x, & -1 \leq x < 0 \\ 0, & 0 \leq x < 1 \\ \tan(x-1) - (x-1), & 1 \leq x < 1 + \frac{\pi}{2} \\ \infty, & 1 + \frac{\pi}{2} \leq x < \infty. \end{cases}$$

In this example the function f is broken into open intervals by the points $(x_0, \ldots, x_5) =$



(a)

(b)

Figure 2.1: (a) $f(x)$ and (b) $\partial f(x)$ from Example 2.1

$(-\infty, -1, 0, 1, 1+\frac{\pi}{2}, \infty)$. Calculating the derivative along each open interval yields:

$$\begin{aligned} f_0'(x) &= undefined, \\ f_1'(x) &= -1, \\ f_2'(x) &= 0, \\ f_3'(x) &= (\tan(x-1))^2, \text{ and} \\ f_4'(x) &= undefined. \end{aligned}$$

At the points x_2 and x_3 the subdifferential values are easily calculated using left and right derivative limits yielding $\partial f(x_2) = [-1, 0]$ and $\partial f(x_3) = \{0\}$. The point x_1 is on the left boundary of the domain of f, and as such is undefined to the left but well defined to the right. It can be seen that all lines with slope less than $\lim_{x \downarrow x_1} f_2'(x) = -1$ are subgradients to f at x_1, thus the subdifferential is given by $\partial f(x_1) = [-\infty, -1]$. Lastly, the point x_4 falls outside the domain of f, and thus has an empty subdifferential. The subdifferential, pictured in Figure 2.1(b), is therefore given by

$$\partial f(x) = \begin{cases} \emptyset, & -\infty < x < -1 \\ [-\infty, -1] & x = -1 \\ \{-1\}, & -1 < x < 0 \\ [-1, 0] & x = 0 \\ \{0\}, & 0 \leq x < 1 \\ \{(\tan(x-1))^2\}, & 1 \leq x < 1 + \frac{\pi}{2} \\ \emptyset, & 1 + \frac{\pi}{2} \leq x < \infty. \end{cases}$$

□

2.3 Symbolic Conjugation in One Dimension

Functions in the class $\mathcal{F}$ are extremely well behaved. Most importantly, they are subdifferentiable on the interior of their domain. Given the subdifferential we may compute the value of the Fenchel conjugate at a point y in two steps:

1. solve $y \in \partial f(x)$ for x, and let $\overline{x}$ be such a solution;

2. use the Fenchel-Young inequality (Theorem 1.24) to obtain $f^*(y) = \overline{x}y - f(\overline{x})$.

The algorithm is most easily illustrated by way of an example.

Example 2.2 Consider the convex function

$$f(x) = \begin{cases} -x, & -\infty < x \leq 0 \\ x^2, & 0 < x < \infty. \end{cases}$$

Calculating the subdifferential results in

$$\partial f(x) = \begin{cases} \{-1\}, & -\infty < x < 0 \\ [-1, 0], & x = 0 \\ \{2x\}, & 0 < x < \infty. \end{cases}$$

We begin by examining the subdifferential over the first open interval $(\infty, 0)$. On this interval the subdifferential takes only one value, namely $y = -1$, and it does so for all x in the open interval. Taking $\overline{x} = -1$ and substituting this into the Fenchel-Young inequality yields $f^*(-1) = (-1)(-1) - f(-1) = 1 - 1 = 0$.

Next we consider the subdifferential at the point $x = 0$. The subdifferential takes on all values $y \in [-1, 0]$. This yields $f^*(y) = (0)(y) - f(0) = 0$ for $y \in [-1, 0]$.

Finally, we consider $\partial f(x)$ over $(0, \infty)$. Inverting $y = \partial f(\overline{x})$ yields $\overline{x} = \frac{1}{2}y$. On this interval, $\partial f(x)$ takes values from $\lim_{x \downarrow 0} 2x = 0$ to $\lim_{x \uparrow \infty} 2x = \infty$. Thus, we find that $f^*(y) = \frac{1}{2}y^2 - f(\frac{1}{2}y) = \frac{1}{4}y^2$ for $y \in (0, \infty)$.

Gluing together these results and defining the conjugate to be infinite elsewhere yields:

$$f^*(y) = \begin{cases} \infty, & -\infty < y < -1 \\ 0, & -1 \leq y \leq 0 \\ \frac{1}{4}y^2, & 0 < y < \infty \end{cases} .$$

□

In general, each piece of a subdifferential falls into one of four categories:

1. $\partial f(x)$ is a constant singleton defined at a point;

2. $\partial f(x)$ is a closed interval defined at a point;

3. $\partial f(x)$ is a constant singleton over an open interval; or,

4. $\partial f(x)$ is a singleton function of x over an open interval.

As illustrated in the example, cases 2 and 4 translate to defining the conjugate $f^*(y)$ over intervals, whereas cases 1 and 3 only define $f^*(y)$ at a single point. Note that cases 1 and 3 essentially contribute no information to the calculation of the conjugate as the conjugate will itself be closed, with the behaviour at these points being implied by lower semi-continuity.

2.4 Function Inversion

In calculating a one dimensional conjugate, the subdifferential must be inverted. However, the subdifferential, while guaranteed non-decreasing on each open interval (x_i, x_{i+1}) and therefore invertible, may be expressed as a piecewise composition of functions that are not trivially invertible over their whole range. This leads to the problem of branch selection in calculating inverses of non-decreasing functions on finite open intervals. We first give a few definitions pertaining to branch points.

Definition 2.3 (Analytic function) Consider a *complex function* $f : \mathbb{C} \to \mathbb{C}$, and let $f(z) = u(x,y) + iv(x,y)$ where $z = x + iy$. If the partial derivatives of f at a point z_0 with respect to x and y are continuous and they satisfy the (Cauchy-Riemann) conditions

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \text{ and } \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y},$$

then the function f is *complex differentiable* at the point z_0. The function f is said to be analytic over a region $R \subset \mathbb{C}$ if it is complex differentiable at every point $z \in R$.

It is worth noting that complex differentiability at a point z_0 is equivalent to having a non-zero radius of convergence for the Taylor series expansion of f about that point. Furthermore, the property of C^∞ on $\mathbb{R}$ is weaker than analyticity. For examples and much more detail refer to [1].

Definition 2.4 (Branch cuts) A branch cut is a curve in the complex plane across which an analytic function is discontinuous.

Consider the function $z \mapsto z^2$. This function is single-valued and maps every input z to a single well-defined value z^2. Its inverse function $\sqrt{z}$, on the other hand, is multi-valued and maps, for example, $1 \mapsto \pm 1$.

Example 2.5 (Branch cut of square root) Consider the point $z = re^{i\theta} \in \mathbb{C}$. Let us adopt the convention that $r \geq 0$, $\theta \in [a, 2\pi + a)$ and $\sqrt{z} = \sqrt{r}e^{i\frac{\theta}{2}}$, where $\sqrt{r}$ is the positive square root of r. Letting $a = \frac{\pi}{2}$ we see:

$$\begin{aligned} \lim_{\theta \downarrow a} \sqrt{z} = {} & e^{i\frac{\pi}{4}} && = i, \\ \lim_{\theta \uparrow 2\pi + a} \sqrt{z} = {} & e^{i\frac{5\pi}{4}} && = -i. \end{aligned}$$

Thus we have a discontinuity occurring across the angle $a = \frac{\pi}{2}$. This discontinuity will exist for any choice of a. In the complex plane, this corresponds to a branch cut departing the origin at an angle a. □

A unique *principal value* can be chosen for multi-valued functions, but the choice can never be made such that the resulting function is continuous over the whole of $\mathbb{C}$. Choosing which is the principal value is largely an issue of convention, and it is usually done to give rise to other simple analytic properties.

For our purposes (inverting multi-valued functions on the real line), a *branch point* is a point on the real line $\mathbb{R}$ at which a branch cut intersects. In Example 2.5, regardless of the convention in force, the single branch point of the square root function is the origin.

Example 2.6 (Simple branch point) We begin with a simple example. Consider the convex function

$$f(x) = \frac{1}{4}x^4, \quad x \in \mathbb{R}.$$

The subdifferential of this function is $\partial f(x) = \{x^3\}$, $x \in \mathbb{R}$. The function $y = x^3$ has three distinct inverses (one for each cube root of unity), given by

$$x \in \left\{ y^{\frac{1}{3}}, \frac{1}{2}\left(-1 + i\sqrt{3}\right) y^{\frac{1}{3}}, \frac{1}{2}\left(-1 - i\sqrt{3}\right) y^{\frac{1}{3}} \right\}.$$

Obviously, the solution we intend is the real cube root. However, for $y < 0$, the value of $y^{\frac{1}{3}}$ is imaginary under the principal branch conventions in force in Maple. Hence, for $y < 0$ another branch must be chosen. In fact, the inverse is found to be

$$x = \begin{cases} \frac{1}{2}(-1 + i\sqrt{3})\, y^{\frac{1}{3}}, & y < 0 \\ y^{\frac{1}{3}}, & 0 \le y \end{cases}.$$

This example demonstrates that in calculating an inverse one may have to select from amongst a finite family of solutions, each being applicable on distinct domains. □

Example 2.7 (Infinite inverses) Consider now the convex function

$$f(x) = \begin{cases} \sin x, & \pi \le x \le 2\pi \\ \infty, & \text{otherwise} \end{cases}.$$

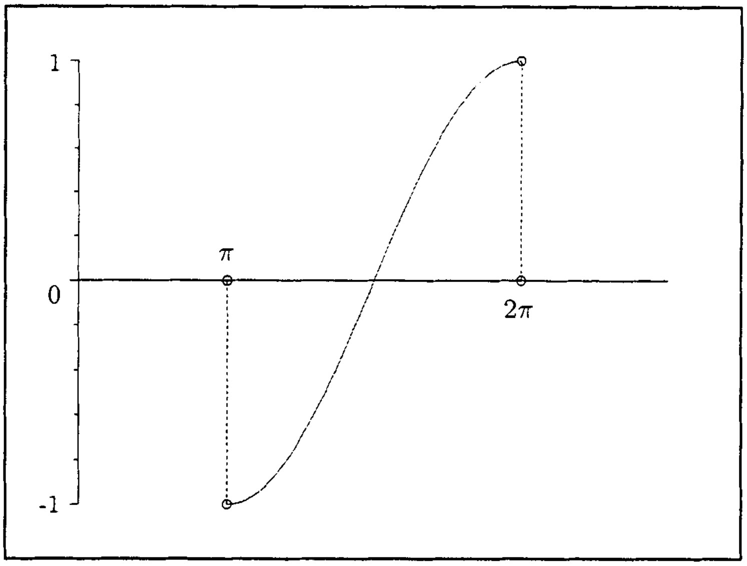


Figure 2.2: Subdifferential of $\sin x$ on $[\pi, 2\pi]$

On the interior of the domain, the subdifferential of this function is calculated as

$$\partial f(x) = \{\cos x\}, \quad \pi < x < 2\pi$$

As seen in Figure 2.2, the subdifferential is increasing and therefore invertible. The general form of the inverse of $y = \cos x$ is

$$x = \arccos y - 2b \arccos y + 2\pi z,$$

where $b \in \mathbb{B} = \{0, 1\}$ and $z \in \mathbb{Z}$. Inspection shows the branch we are interested in is characterized by $b = z = 1$, yielding an inverse of

$$x = -\arccos y + 2\pi.$$

This example illustrates the possibility of having to choose an inverse from amongst an infinite family of solutions. □

In the most general case, there may be the need to choose inverses from a finite collection of infinite families of inverses, with multiple distinct solutions over disjoint sub-intervals.

The first problem that must be solved is that of finding the boundaries (branch points) between intervals over which different branches may apply. We appeal first to a result from elementary complex analysis.

Theorem 2.8 ([1], Chapter 3, Theorem 11, page 131) *Suppose that $f(z)$ is analytic at z_0, $f(z_0) = w_0$, and that $f(z) - w_0$ has a zero of order n at z_0. If $\epsilon > 0$ is sufficiently small, there exists a corresponding $\delta > 0$ such that for all a with $|a - w_0| < \delta$ the equation $f(z) = a$ has exactly n roots in the disk $|z - z_0| < \epsilon$.*

Proof: The proof of this theorem is beyond the scope of this thesis. For full details, refer to [1]. ■

Corollary 2.9 (Location of branch points) *Suppose that f is as in Theorem 2.8. Suppose furthermore that $f(z)$ is analytic on the entire neighborhood $|z - z_0| < \epsilon$, and let $g_1(a), \ldots, g_n(a)$ represent the n roots of $f(z) = a$ on the neighborhood $|a - w_0| < \delta$. Then $g_1(w_0) = \cdots = g_n(w_0) = z_0$.*

Proof: Due to the n^{th} order zero of $f(z)$ at z_0, it follows that $f(z)$ may be expressed as $f(z) - w_0 = (z - z_0)^n g(z)$, where $g(z) \neq 0$, for all z with $|z - z_0| < \epsilon$. Due to the analyticity of $f(z)$ and the existence of exactly n roots by Theorem 2.8, for any a with $|a - w_0| < \delta$ we can write $f(z) - a = (z - g_1(a)) \cdots (z - g_n(a))h(z)$, for some $h(z) \neq 0$. Since $\lim_{a \to z_0} f(z) - a = f(z) - w_0$, it follows that $\lim_{a \to z_0} (z - g_1(a)) \cdots (z - g_n(a))h(z) = (z - z_0)^n g(z)$, and therefore $(z - g_1(w_0)) \cdots (z - g_n(w_0))h(z) = (z - z_0)^n g(z)$. Suppose $g_i(w_0) \neq z_0$ for some i. Then, since $h(z_0) \neq 0$, it follows that the left hand side of the equation has at most $n - 1$ roots at w_0, a contradiction. Thus, it must be that $g_1(w_0) = \cdots = g_n(w_0) = z_0$. ■

Corollary 2.9 tells us that anywhere a function has n inverses, the branches are equal at a point with a zero of order n. This tells us that points where two branches are equal occur at zeroes of the first derivative. Thus, when wanting to determine the inverse of a function f over the interval (a, b) we first find all solutions to $f'(x) = 0$, for $x \in (a, b)$. If we can find all of the zeroes then we are guaranteed to have found all of the possible branch points, and can proceed to find the unique branch which is the inverse over each disjoint sub-interval. In order for Corollary 2.9 to apply for our algorithms we need to restrict ourselves to input functions that are *real analytic*; in other words, functions f that are analytic on dom $f \subset \mathbb{R}$.

Example 2.10 (Branch points) Consider $y = x^3,\ x \in \mathbb{R}$ from Example 2.6. Taking the derivative yields $y' = 3x^2,\ x \in \mathbb{R}$. Solving $3x^2 = 0$ yields the single solution $b_1 = 0$. Thus, we are assured that the inverse of x^3 along the real line has at most one branch point, located at the origin. □

Once the domain of the function has been partitioned into disjoint sub-intervals the inverses over each of these may be determined. This can be accomplished by testing each possible inverse in $\mathcal{G}$ over each distinct interval. When determining the inverses, there are two cases to consider as outlined in Examples 2.6 and 2.7.

2.5 Numerical Methods

Often, there will exist no closed-form conjugate and in order to gain any insight into the nature of a subdifferential or conjugate we must resort to numerical methods.

Although the Legendre-Fenchel transform is fundamental in convex analysis, until relatively recently no algorithms were available to compute it efficiently. Early algorithms were aimed at solving Hamilton-Jacobi equations [7] or Burger's equation [14]. These algorithms were designed similarly to Fast Fourier Transforms, and could compute an m point conjugate to a function evaluated at n points in $O((n+m)\log(n+m))$ time. More recent work by Yves Lucet in [11] and [12] describes an algorithm (the linear time Legendre transform, or *LLT*) that runs in $O(n+m)$ time. The key innovation of this algorithm exploits the already sorted nature of an array of function evaluation points.

Consider a one dimensional function $f : \mathbb{R} \to \overline{\mathbb{R}}$ restricted to a closed finite interval $[a,b]$. The restricted function $f_{[a,b]}$ is defined as $f + \delta_{[a,b]}$. Given a sorted set of points $X = \{x_1, \ldots, x_n\}$ we define the discrete approximation f_X of f as $f_X(x) = f(x)$ for $x \in X$, with linear interpolation defining f_X at points not in X. We quote a result from [7] and [11].

Proposition 2.11 (Convergence of discrete Legendre transform) *Let $X = \{x_1, \ldots, x_n\}$ be a subset of $[a,b]$ such that for all $y \in [a,b]$ there exists $x_i \in X$ with $|x_i - y| \le (b-a)/n$. Let f be a function from $\mathbb{R} \to \overline{\mathbb{R}}$.*

1. *If f is upper semi-continuous on a neighborhood of $[a,b]$, then $(f_X)^*$ converges pointwise to $(f_{[a,b]})^*$.*

2. *If f is twice continuously differentiable on a neighborhood of $[a,b]$, then for all y*

$$0 \leq (f_{[a,b]})^*(y) - (f_X)^*(y) \leq \frac{(b-a)^2}{2n^2} \max_{x \in [a,b]} f''(x).$$

The above result states that the conjugate of the discrete approximation of f converges pointwise to the conjugate of the restriction f to $[a,b]$. As for convergence of the conjugate of the restriction towards the conjugate, we have a much stronger result provided by Hiriart-Urruty in [9]: $(f_{[-a,a]})^* = f^*$ for sufficiently large a.

Proposition 2.12 (Convergence of restricted functions)
The following are equivalent:

1. *$\partial f^*(y) \cap [-a,a] \neq \emptyset$; and,*
2. *$(f_{[-a,a]})^*(y) = f^*(y)$*

Combining the previous two results tells us that the discrete approximation to the conjugate will converge pointwise to the actual conjugate as we increase the range and number of evaluation points.

2.5.1 The Linear-time Legendre Transform

The problem is to compute $(f_X(y))^* = g_Y(y)$ for y in $Y = \{y_1, \ldots, y_m\}$, where $x_1 < \ldots < x_n$ and $y_1 < \ldots < y_m$. Suppose f is convex. Then we can use the monotonicity of the subdifferential ∂f more efficiently than other algorithms by introducing the (increasing) sequence of slopes

$$s_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}.$$

Since f is convex finding the support point of the minorizing line with a slope y is rather straight-forward (we can perform a search through the increasing s_i), and together with Equation 1.22 yields the value of the discrete conjugate g_Y at y as:

1. if $y < s_1$, then $g_Y(y) = yx_1 - f(x_1)$;
2. if $y > s_{n-1}$, then $g_Y(y) = yx_n - f(x_n)$; and,
3. if $s_{i-1} < y \leq s_i$, then $g_Y(y) = yx_i - f(x_i)$.

The above logic is assuming that f and hence f_X is convex. Since the set X is sorted, we may apply any linear time algorithm (see [10] or [15]) to first calculate the convex hull of f_X.

Thus, given f_X, X and Y the entire algorithm can be described as follows:

1. compute $\bar{f}_X$, the convex hull of f_X;
2. compute slopes $S = \{s_1, \ldots, s_{n-1}\}$; and,
3. for each y_j compute $g_Y(y_j)$ by finding the index i such that $s_{i-1} < y_j \leq s_i$.

Assuming X is stored in sorted order, the first two steps are $O(n)$. Since both Y and S are in increasing order, the last step can be done in a single $O(n+m)$ loop. Thus, the entire algorithm runs in $O(n+m)$ time. For further details of algorithmic performance, refer to [11] and [12].

Chapter 3

Convex Analysis in Higher Dimensions

Recall the definition of the Fenchel conjugate from (Section 1.6). In higher dimensions this can be rewritten as:

$$\begin{aligned} f^*(\mathbf{y}) &= \sup_{\mathbf{x}} \{\langle \mathbf{x}, \mathbf{y}\rangle - f(\mathbf{x})\} \\ &= \sup_{x_1,\ldots,x_n} \left\{ \sum_{i=1}^{n} x_i y_i - f(\mathbf{x}) \right\} \\ &= \sup_{x_1} \left\{ x_1 y_1 + \sup_{x_2} \left\{ x_2 y_2 + \cdots + \sup_{x_n} \{x_n y_n - f(\mathbf{x})\} \cdots \right\} \right\}. \end{aligned}$$

We introduce the concept of a *partial conjugate*. Consider an n dimensional function that has had a one dimensional conjugate calculated with respect to the variable x_i. The notation f^{x_i} then represents this partial conjugate of f with respect x_i. The above may be rewritten as

$$f^* = (-(\cdots - (f^{x_n} \cdots)^{x_2})^{x_1}.$$

This is equivalent to taking the conjugate along the x_n variable, negating the result, taking the conjugate along the x_{n-1} variable, negating the result, etc, until the conjugate is finally taken along the x_1 variable. In other words, the conjugate of an n dimensional function can be calculated as a sequence of n iterated one dimensional conjugates. While the concept of *iterated conjugation* is straight-forward, various complications arise in practice which must be addressed.

The notion of iterated conjugation can be likened in many respects to that of iterated integration, the standard technique used for calculating multiple integrals. In fact, as will be shown in Section 3.3.3, the necessary juggling of partial conjugates between conjugation iterations can be equated directly to the problem of changing the variable order in a multiple integral.

Note that in the special case where the function f is *separable* the conjugate may be simplified to

$$\begin{aligned} f^*(\mathbf{y}) &= \sup_{\mathbf{x}} \sum_{i=1}^{n} (x_i y_i - f(x_i)) \\ &= \sum_{i=1}^{n} \left(\sup_{x_i} \{x_i y_i - f(x_i)\} \right) \\ &= \sum_{i=1}^{n} f^*(y_i), \end{aligned}$$

which allows us to calculate the conjugate as n separate one dimensional conjugates. However, this is not usually the case.

3.1 A Good Class of Functions

The natural space to work in is the recursive extension to $\mathcal{F}$. An n dimensional function f is in $\mathcal{F}^n$ if:

(i) $f(x_1, \ldots, x_n)$ is a function from $\mathbb{R}^n$ to $\overline{\mathbb{R}}$;

(ii) $f(x_1, \ldots, x_n)$ is a closed convex function;

(iii) $f(x_1, \ldots, x_n)$ is continuous on its effective domain; and,

(iv) there are finitely many points a_i such that $a_0 = -\infty < a_1 < \cdots < a_{m-1} < a_m = \infty$ and f restricted to each open interval (a_i, a_{i+1}) is in $\mathcal{F}^{n-1}$ (where $\mathcal{F}^1 = \mathcal{F}$) with respect to the variables $x_2, \ldots, x_n$.

Consider a function $f(x_1, \ldots, x_n) \in \mathcal{F}^n$ defined on each open interval (a_{i_1}, a_{i_1+1}) as f_{i_1}. Each f_{i_1} is similarly defined on the open interval $(a_{(i_1,i_2)}, a_{(i_1,i_2+1)})$ as $f_{(i_1,i_2)}$. Taking this to its extreme, on the n dimensional open interval
$(a_{i_1}, a_{i_1+1}) \times (a_{(i_1,i_2)}, a_{(i_1,i_2+1)}) \times \cdots \times (a_{(i_1,\ldots,i_{n-1})}, a_{(i_1,\ldots,i_{n-1}+1)})$, f is given by $f_{(i_1,\ldots,i_{n-1})}$.

The space $\mathcal{F}^n$ is very well suited to our purpose as it allows a relatively compact representation of any piecewise continuous convex function defined on $\mathbb{R}^n$. It is also recursive in nature, and therefore naturally representable and manipulated in a computer algebra system.

3.2 One dimensional Conjugation With Bounded Parameters

In order to calculate the conjugate of a function f in $\mathcal{F}^n$ we must first calculate the one dimensional conjugate of each $f_{(i_1,\ldots,i_{n-1})}$. Each $f_{(i_1,\ldots,i_{n-1})}$ may be dependent upon the variables $x_1, \ldots, x_{n-1}$. Thus, in calculating the partial conjugate with respect to x_n, the variables $x_1, \ldots, x_{n-1}$ must be treated as real parameters with bounds $a_{(i_1,\ldots,i_k)} < x_k < a_{(i_1,\ldots,i_{k+1})}$. The existence of these bounded parameters makes it more difficult to choose the appropriate branch when inverting the subdifferential.

Example 3.1 (Inversion with bounded parameters) Consider $f(x_2) = x_1 x_2^3$ defined for $x_2 > 0$. Inverting this with respect to x_2 yields 3 possible solutions:

$$f^{-1}(y_2) \in \left\{ \frac{1}{x_1} \left(y_2 x_1^2\right)^{\left(\frac{1}{3}\right)}, \frac{1}{2x_1} \left(y_2 x_1^2\right)^{\left(\frac{1}{3}\right)} \left(-1 + i\sqrt{3}\right), \frac{-1}{2x_1} \left(y_2 x_1^2\right)^{\left(\frac{1}{3}\right)} \left(1 + i\sqrt{3}\right) \right\}$$

If $x_1 < 0$ then

$$f^{-1}(y_2) = \frac{-1}{2x_1} \left(y_2 x_1^2\right)^{\left(\frac{1}{3}\right)} \left(1 + i\sqrt{3}\right).$$

However, if $x_1 > 0$ then

$$f^{-1}(y_2) = \frac{1}{x_1} \left(y_2 x_1^2\right)^{\left(\frac{1}{3}\right)}.$$

Thus, the knowledge of any free-parameter bounds is required in order to make the correct decision in calculating the one dimensional conjugates. In practice, we supply any given □

3.3 Variable Reordering

Functions defined in $\mathcal{F}^n$ have an implicit variable order due to their recursive structure. A function $f \in \mathcal{F}^n$ defined with the variable order $x_1, x_2, \ldots, x_n$ may only have the partial conjugate calculated along the x_n variable, at which point the variables of the new partially conjugated function are $x_1, \ldots, x_{n-1}, y_n$. For this function to be conjugated along any other variable, it must first have its variables reordered so that one of $x_1, \ldots, x_{n-1}$ is the last variable. We illustrate with an example in $\mathcal{F}^2$.

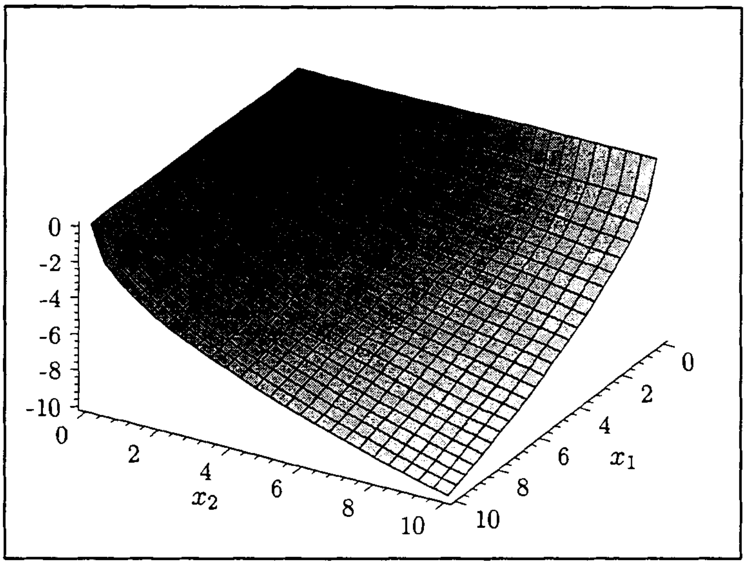


Figure 3.1: $f(x_1, x_2)$ from Example 3.2

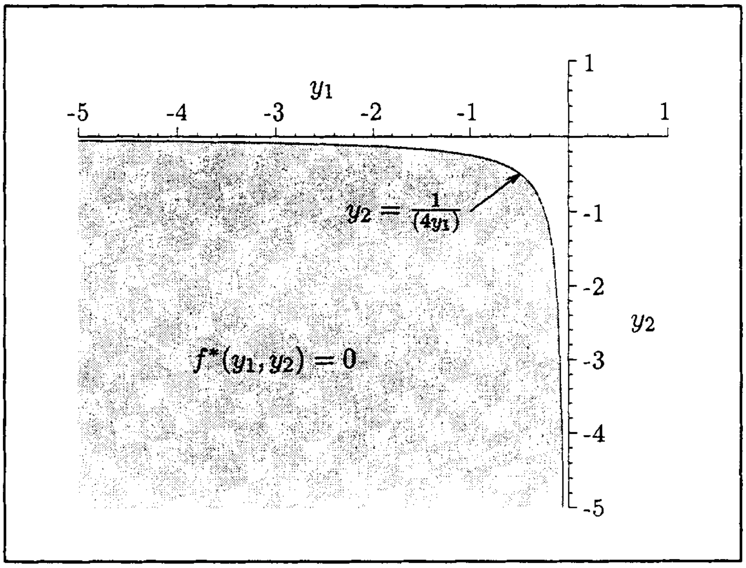


Figure 3.2: A plan view of $f^*(y_1, y_2)$ from Example 3.2

Example 3.2 (Product of roots) Consider the two dimensional function (shown in Figure 3.1):

$$f(x_1,x_2) = \begin{cases} \begin{cases} \infty, & \forall x_2 \end{cases}, & x_1 < 0 \\ \begin{cases} \infty, & x_2 < 0 \\ 0, & x_2 = 0 \\ 0, & 0 < x_2 \end{cases}, & x_1 = 0 \\ \begin{cases} \infty, & x_2 < 0 \\ 0, & x_2 = 0 \\ -\sqrt{x_1 x_2}, & 0 < x_2 \end{cases}, & 0 < x_1 \end{cases}$$

Calculating the partial conjugate with respect to the x_2 axis involves calculating two one dimensional partial conjugates; one along the line $x_1 = 0$ and the other over the half-plane $0 < x_1$. Calculating these conjugates (and negating the results) yields:

$$f^{x_2}(x_1,y_2) = \begin{cases} \begin{cases} \infty, & \forall y_2 \end{cases}, & x_1 < 0 \\ \begin{cases} 0, & y_2 < 0 \\ 0, & y_2 = 0 \\ \infty, & 0 < y_2 \end{cases}, & x_1 = 0 \\ \begin{cases} \frac{x_1}{4y_2}, & y_2 < 0 \\ \infty, & y_2 = 0 \\ \infty, & 0 < y_2 \end{cases}, & 0 < x_1 \end{cases}$$

We now wish to calculate the partial conjugate along the x_1 variable in order to complete the two dimensional conjugation. However, in order to do this, we must first reorder the variables to (y_2, x_1). In this example this is easily done through inspection, resulting in:

$$f^{x_2}(y_2,x_1) = \begin{cases} \begin{cases} \infty, & x_1 < 0 \\ 0, & x_1 = 0 \\ \frac{x_1}{4y_2}, & 0 < x_1 \end{cases}, & y_2 < 0 \\ \begin{cases} \infty, & \forall x_1 \end{cases}, & y_2 = 0 \\ \begin{cases} \infty, & \forall x_1 \end{cases}, & 0 < y_2 \end{cases}$$

We may now proceed to calculate the complete conjugate by partially conjugating along the x_1 axis. There are two distinct one dimensional conjugates to be calculated along the line $y_2 = 0$ and the half-plane $y_2 < 0$. This yields:

$$f^*(y_2, y_1) = \begin{cases} \begin{cases} 0, & y_1 < \frac{1}{4y_2} \\ 0, & y_1 = \frac{1}{4y_2} \\ \infty, & \frac{1}{4y_2} < y_1 \end{cases}, & y_2 < 0 \\ \left\{ \infty, \ \forall y_1 \right. , & y_2 = 0 \\ \left\{ \infty, \ \forall y_1 \right. , & 0 < y_2 \end{cases}$$

It is desirable to have the conjugated function in the same variable order as the original function. This involves yet another variable reordering to (y_1, y_2). The result of this operation is the final conjugate:

$$f^*(y_1, y_2) = \begin{cases} \begin{cases} 0, & y_2 < \frac{1}{4y_1} \\ 0, & y_2 = \frac{1}{4y_1} \\ \infty, & \frac{1}{4y_1} < y_2 \end{cases}, & y_1 < 0 \\ \left\{ \infty, \ \forall y_2 \right. , & y_1 = 0 \\ \left\{ \infty, \ \forall y_2 \right. , & 0 < y_1 \end{cases}$$

The conjugate is easily visualized as the indicator function of a convex set in $\mathbb{R}^2$, this set being illustrated in Figure 3.2. □

To simplify the requirements of variable reordering, we introduce the notion of *pivoting*. A pivot is a change of variable order from $(x_1, \ldots, x_n)$ to $(x_n, x_1, \ldots, x_{n-1})$. Simply stated, the penultimate variable becomes the last, and the last becomes the first. Pivoting is a constrained form of general variable reordering, but it is sufficient to perform conjugate calculations. To further simplify the discussion of pivoting, we first change the space of the problem.

3.3.1 Region Representation

A function in $\mathcal{F}^n$ can be thought of as being defined by a collection of regions, where a region r is a pair consisting of a set $S \subseteq \mathbb{R}^n$ and a function f which is continuously differentiable over S. The set S is defined as $S = \{\mathbf{x} : x_1 \in X_1, x_2 \in X_2, \ldots, x_n \in X_n\}$, where the one dimensional sets X_i are either open intervals $(a_i(x_1, \ldots, x_{i-1}), b_i(x_1, \ldots, x_{i-1}))$ or singletons $\{a_i(x_1, \ldots, x_{i-1})\}$. For instance, the function in Example 3.2 may be represented by the following collection of regions:

$$\begin{array}{llllll}
(\{\mathbf{x} & : & x_1 \in (-\infty, 0), & x_2 \in \mathbb{R} & \}, & \infty &), \\
(\{\mathbf{x} & : & x_1 = 0, & x_2 \in (-\infty, 0) & \}, & \infty &), \\
(\{\mathbf{x} & : & x_1 = 0, & x_2 = 0 & \}, & 0 &), \\
(\{\mathbf{x} & : & x_1 = 0, & x_2 \in (0, \infty) & \}, & 0 &), \\
(\{\mathbf{x} & : & x_1 \in (0, \infty), & x_2 \in (-\infty, 0) & \}, & \infty &), \\
(\{\mathbf{x} & : & x_1 \in (0, \infty), & x_2 = 0 & \}, & 0 &) \text{ and} \\
(\{\mathbf{x} & : & x_1 \in (0, \infty),. & x_2 \in (0, \infty) & \}, & -\sqrt{x_1 x_2} &).
\end{array}$$

For the sake of compactness, we may ignore any region over which the function is identically ∞, and make that value implicit for any point $\mathbf{x} \in \mathbb{R}^n$ that does not fall within one of the defined regions. The conversion of a function in $\mathcal{F}^n$ to a collection of regions is a straightforward recursive process. The reverse process is also possible, but much greater care need be taken.

3.3.2 Region Representation to Recursive Representation

Consider a collection of regions $\mathcal{R} = \{r_i\}$ where $r_i = (\{x_1 \in X_{i,1}, \ldots,\ x_n \in X_{i,n}\}, f_i)$. Partition $\mathcal{R}$ into two sets, $\mathcal{R}_S$ and $\mathcal{R}_I$, where

$$\mathcal{R}_S = \{r_i : X_{i,1} \text{ is a singleton}\}, \text{ and}$$
$$\mathcal{R}_I = \{r_i : X_{i,1} \text{ is an open interval}\}.$$

Let $\mathcal{E}$ be the collection of interval end-points and singletons (as applicable) along the first dimension of each region:

$$\mathcal{E} = \{-\infty, \infty\} \cup \{a_{i,1} : r_i \in \mathcal{R}_S\} \cup \{a_{i,1}, b_{i,1} : r_i \in \mathcal{R}_I\}.$$

Let the points in $\mathcal{E}$ be indexed by e_i, where $-\infty = e_0 < e_2 < \cdots < e_m = \infty$. We construct a function $g \in \mathcal{F}^n$ defined over the finitely many points e_i. Over each open interval (e_i, e_{i+1}) we define g as $g_{(i,i+1)} \in \mathcal{F}^{n-1}$, and at each point e_i we define g as $g_i \in \mathcal{F}^{n-1}$. We construct $g_{(i,i+1)}$ as follows:

(i) Let $\overline{\mathcal{R}}$ be the collection of all regions in $\mathcal{R}_I$ that overlap the interval (e_i, e_{i+1}), reduced to $n-1$ dimensions by removing the constraint along the first dimension:

$$\overline{\mathcal{R}} = \{(\{x_2 \in X_{j,2}, \ldots,\ x_n \in X_{j,n}\}, f_j) : r_j \in \mathcal{R}_I,\ X_{j,1} \cap (e_i, e_{i+1}) \neq \emptyset\}.$$

(ii) Let $g_{(i,i+1)}$ be the function in $\mathcal{F}^{n-1}$ returned by recursively processing the regions $\overline{\mathcal{R}}$ using the procedure just described.

Similarly, we may construct each g_i by recursively processing the set of $(n-1)$ dimensional regions given by:

$$\overline{\mathcal{R}} = \{(\{x_2 \in X_{j,2}, \ldots,\ x_n \in X_{j,n}\}, f_j) : r_j \in \mathcal{R},\ e_i \in X_{j,1}\}.$$

The final case to consider is the base case, when the dimension has been reduced to 1. In this case, region representation and recursive representation are much the same thing and we can directly equate the two. Consider $f \in \mathcal{F}$ in region representation as:

$$\begin{array}{llll}
(\{x \in (-\infty, a_1) & \}, f_1 &), & \\
(\{x = a_1 & \}, f_2 &), & \\
(\{x \in (a_1, a_2) & \}, f_3 &), & \\
\quad\vdots & \quad\vdots & & \\
(\{x \in (a_{m-2}, a_{m-1}) & \}, f_{2(m-1)-1} &), & \\
(\{x = a_{m-1} & \}, f_{2(m-1)} &), & \text{and} \\
(\{x \in (a_{m-1}, \infty) & \}, f_{2m-1} &). &
\end{array}$$

This is equivalent in recursive representation to:

$$f = \begin{cases} f_1, & x \in (-\infty, a_1) \\ f_2, & x = a_1 \\ \quad\vdots & \quad\vdots \\ f_{2(m-1)}, & x = a_{m-1} \\ f_{2m-1}, & x \in (a_{m-1}, \infty) \end{cases}$$

The entire procedure is clarified in the following example.

Example 3.3 Consider the following set of regions:

$$\begin{array}{rlllcl}
r_1 = & (\{\mathbf{x} : x_1 = -1, & x_2 = -1 & \}, & 0 &) \\
r_2 = & (\{\mathbf{x} : x_1 = -1, & x_2 = 0 & \}, & 0 &) \\
r_3 = & (\{\mathbf{x} : x_1 = -1, & x_2 = 1 & \}, & 1 &) \\
r_4 = & (\{\mathbf{x} : x_1 = 0, & x_2 = 1 & \}, & 1 &) \\
r_5 = & (\{\mathbf{x} : x_1 = 1, & x_2 = -1 & \}, & 0 &) \\
r_6 = & (\{\mathbf{x} : x_1 = 1, & x_2 = 0 & \}, & 0 &) \\
r_7 = & (\{\mathbf{x} : x_1 = 1, & x_2 = 1 & \}, & 2 &) \\
r_8 = & (\{\mathbf{x} : x_1 = -1, & x_2 \in (-1,0) & \}, & 0 &) \\
r_9 = & (\{\mathbf{x} : x_1 = -1, & x_2 \in (0,1) & \}, & x_2 &) \\
r_{10} = & (\{\mathbf{x} : x_1 = 0, & x_2 \in (0,1) & \}, & x_2 &) \\
r_{11} = & (\{\mathbf{x} : x_1 = 1, & x_2 \in (-1,0) & \}, & 0 &) \\
r_{12} = & (\{\mathbf{x} : x_1 = 1, & x_2 \in (0,1) & \}, & 2x_2 &) \\
r_{13} = & (\{\mathbf{x} : x_1 \in (-1,0), & x_2 = 1 & \}, & 1 &) \\
r_{14} = & (\{\mathbf{x} : x_1 \in (-1,1), & x_2 = -1 & \}, & 0 &) \\
r_{15} = & (\{\mathbf{x} : x_1 \in (-1,1), & x_2 = 0 & \}, & 0 &) \\
r_{16} = & (\{\mathbf{x} : x_1 \in (0,1), & x_2 = 1 & \}, & (1+x_1) &) \\
r_{17} = & (\{\mathbf{x} : x_1 \in (-1,0), & x_2 \in (0,1) & \}, & x_2 &) \\
r_{18} = & (\{\mathbf{x} : x_1 \in (0,1), & x_2 \in (0,1) & \}, & (1+x_1)x_2 &) \\
r_{19} = & (\{\mathbf{x} : x_1 \in (-1,1), & x_2 \in (-1,0) & \}, & 0 &)
\end{array}$$

We first partition these regions into the two sets $\mathcal{R}_S = \{r_i : i = 1, \ldots, 12\}$ and $\mathcal{R}_S = \{r_i : i = 13, \ldots, 19\}$. Extracting the end-points and singletons yields the set $\mathcal{E} = \{-\infty, -1, 0, 1, \infty\}$.

We begin with the first open interval $(-\infty, -1)$. Since there are no regions that define the function over this interval we can infer that $g_{(0,1)} = \infty$, $\forall x_2$.

Consider the point $e_1 = -1$. We determine that each of $\{r_1, r_2, r_3, r_8, r_9\}$ contain this point. Removing the first dimension from these regions yields:

$$\overline{\mathcal{R}} = \{(\{x_2 = -1\}, 0), (\{x_2 \in (-1,0)\}, 0), (\{x_2 = 0\}, 0), (\{x_2 \in (0,1)\}, x_2), (\{x_2 = 1\}, 1)\}.$$

This is a one dimensional region representation which is converted to recursive form as:

$$g_1 = \begin{cases} \infty, & x_2 \in (-\infty, -1) \\ 0, & x_2 = -1 \\ 0, & x_2 \in (-1, 0) \\ 0, & x_2 = 0 \\ x_2, & x_2 \in (0, 1) \\ 1, & x_2 = 1 \\ \infty, & x_2 \in (1, \infty) \end{cases}$$

Next we process the open interval $(e_1, e_2) = (-1, 0)$. We find that the regions $\{r_{13}, r_{14}, r_{15}, r_{17}, r_{19}\}$ overlap with this interval. Reducing by one dimension yields:

$$\overline{\mathcal{R}} = \{(\{x_2 = -1\}, 0), (\{x_2 \in (-1,0)\}, 0), (\{x_2 = 0\}, 0), (\{x_2 \in (0,1)\}, x_2), (\{x_2 = 1\}, 1)\}.$$

Constructing the appropriate $g_{(1,2)} \in \mathcal{F}$ yields that $g_{(1,2)} = g_1$.

We may proceed similarly with calculations for g_3, $g_{(3,4)}$, g_4 and $g_{(4,5)}$. Finally, we construct $g \in \mathcal{F}^2$ as:

$$g = \begin{cases} \begin{cases} \infty, & x_2 \in (-\infty, \infty) \end{cases}, & x_1 \in (-\infty, -1) \\ \begin{cases} \infty, & x_2 \in (-\infty, -1) \\ 0, & x_2 \in [-1, 0] \\ x_2, & x_2 \in (0, 1] \\ \infty, & x_2 \in (1, \infty) \end{cases}, & x_1 \in [-1, 0] \\ \begin{cases} \infty, & x_2 \in (-\infty, -1) \\ 0, & x_2 \in [-1, 0] \\ (1 + x_1)x_2, & x_2 \in (0, 1] \\ \infty, & x_2 \in (1, \infty) \end{cases}, & x_1 \in (0, 1] \\ \begin{cases} \infty, & x_2 \in (-\infty, \infty) \end{cases}, & x_1 \in (1, \infty) \end{cases}$$

□

With the ability to convert easily between recursive representation and region representation, we may pivot a function in recursive representation by first converting to region representation, pivoting the individual regions, and then converting back to recursive representation.

3.3.3 Region Pivoting

Consider a region $r = (S, f)$, with $S = \{\mathbf{x} : x_1 \in X_1,\ x_2 \in X_2, \ldots,\ x_n \in X_n\}$. Let $V \subseteq \{x_1, \ldots, x_{n-1}\}$ be the set of variables upon which X_n is dependent.

If $V = \emptyset$, then pivoting the set is as simple as rewriting it in the pivoted order $\overline{S} = \{\mathbf{x} : x_n \in X_n,\ x_1 \in X_1, \ldots,\ x_{n-1} \in X_{n-1}\}$, as X_n is independent of any previous variables.

If $|V| = 1$, then let $V = \{x_k\}$. We may pivot the two dimensional set $\{(x_k, x_n) : x_k \in X_k,\ x_n \in X_n\}$ yielding $\{(x_n, x_k) : x_k \in \overline{X}_n,\ x_n \in \overline{X}_k\}$, and thereby pivot S as $\overline{S} = \{\mathbf{x} : x_n \in \overline{X}_n,\ x_1 \in X_1, \ldots,\ x_k \in \overline{X}_k, \ldots,\ x_{n-1} \in X_{n-1}\}$. We discuss two dimensional set pivoting in greater detail in the subsequent section. If $|V| > 1$ the problem becomes much more difficult, and no general solution is currently known.

As discussed earlier, the operation of changing variable order of a function in $\mathcal{F}^n$ is completely analogous to that of changing the order of integration in a multiple-integral. Consider the integral

$$\int_S f(\mathbf{x}) d\mathbf{x}.$$

This may be rewritten as the multiple integral

$$\int_{X_1} \cdots \int_{X_n} f(\mathbf{x}) dx_n \cdots dx_1.$$

Changing the order of integration to

$$\int_{X_n} \int_{X_1} \cdots \int_{X_{n-1}} f(\mathbf{x}) dx_{n-1} \cdots dx_1 dx_n$$

is an equivalent operation to pivoting the original domain S. Consequently, all of the techniques discussed in this section may be applied directly to this problem as well.

3.3.4 Region Pivoting in Two Dimensions

Consider the set $S = \{(x_1, x_2) : x_1 \in X_1,\ x_2 \in X_2(x_1)\}$. Since X_2 is dependent on x_1, it follows that X_1 can not be a singleton, and is therefore an open interval (a, b). If X_2 is an open interval, let $X_2(x_1) = (f(x_1), g(x_1))$. For further generality, if X_2 is a singleton at every x_1, let $X_2(x_1) = \{f(x_1)\}$ and define $g = f$.

Pivoting a two dimensional region will involve inverting f and g. However, f and g may not be monotonic over the interval (a, b) and therefore may not have an inverse. Similarly, as in Section 2.4 there may be branch points in (a, b).

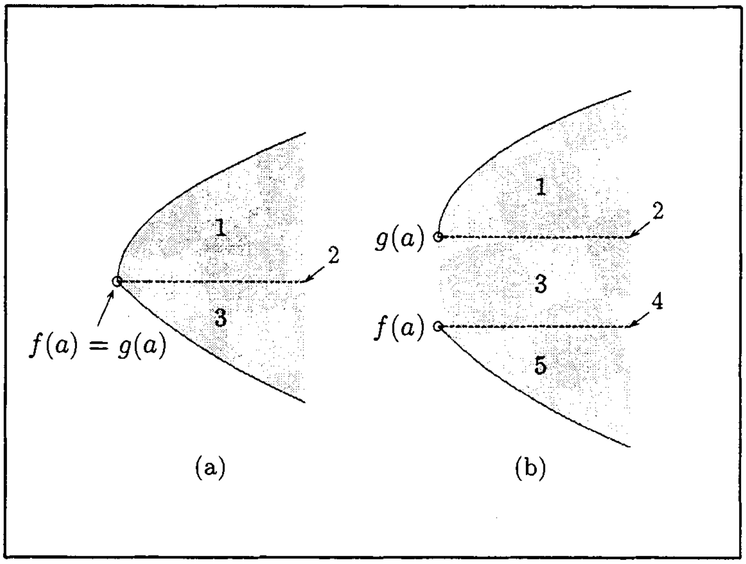


Figure 3.3: Pivoting two monotone regions

Prior to pivoting S, we first split S into a collection of disjoint sets, if necessary. Let $C_f = \{x_1 : f'(x_1) = 0, x_1 \in (a,b)\}$ if f is non-constant, and $C_f = \emptyset$ otherwise. Let $a = c_0 < c_1 < \cdots < c_{m+1} = b$, where $C_f \cup C_g = \{c_i : i \in \{1,\ldots,m\}\}$. By splitting the region at every possible branch point we have assured that f and g have unique inverses over each interval (c_i, c_{i+1}). We may therefore partition S into a collection of disjoint sets $S = S_1 \cup P_1 \cup \cdots \cup P_{m-1} \cup S_m$, where $S_i = \{(x_1,x_2) : x_1 \in (c_i, c_{i+1}),\ x_2 \in X_2(x_1)\}$, and $P_i = \{(x_1,x_2) : x_1 = c_i,\ x_2 \in X_2(c_i)\}$.

Without loss of generality, we may now assume that f and g are either constant or strictly monotonic on (a,b). Suppose X_2 is a singleton, and therefore $f = g$. In this case f is either strictly increasing or decreasing. If strictly increasing, we may pivot S as $\overline{S} = \{(x_2,x_1) : x_2 \in (f(a),f(b)),\ x_1 = f^{-1}(x_2)\}$. If f is strictly decreasing, this becomes $\overline{S} = \{(x_2,x_1) : x_2 \in (f(b),f(a)),\ x_1 = f^{-1}(x_2)\}$. In the rest of the cases, $f \neq g$.

Consider now the case when f is strictly decreasing and g is strictly increasing. Restricting ourselves to $f(x) \leq g(x)$, there are two sub-cases to consider: either $f(a) = g(a)$, or $f(a) < g(a)$ (where $f(a)$ implies $\lim_{x_1 \to a} f(x_1)$ when $a = \pm\infty$). These two cases are illustrated in Figure 3.3. In the first sub-case we may pivot S by splitting it into the following

3 disjoint sets:

$$\begin{aligned}
\overline{S}_1 &= \{(x_2,x_1) \ : \ x_2 \in (f(a),g(b)), \quad x_1 \in (a,g^{-1}(x_2)) \ \}, \\
\overline{S}_2 &= \{(x_2,x_1) \ : \ x_2 = f(a), \quad x_1 \in (a,b) \ \} \text{ and} \\
\overline{S}_3 &= \{(x_2,x_1) \ : \ x_2 \in (f(b),f(a)), \quad x_1 \in (f^{-1}(x_2),b) \ \}.
\end{aligned}$$

In the second sub-case we may pivot S by splitting it into the following 5 disjoint sets:

$$\begin{aligned}
\overline{S}_1 &= \{(x_2,x_1) \ : \ x_2 \in (g(a),g(b)), \quad x_1 \in (a,g^{-1}(x_2)) \ \}, \\
\overline{S}_2 &= \{(x_2,x_1) \ : \ x_2 = g(a), \quad x_1 \in (a,b) \ \}, \\
\overline{S}_3 &= \{(x_2,x_1) \ : \ x_2 \in (f(a),g(a)), \quad x_1 \in (a,b) \ \}, \\
\overline{S}_4 &= \{(x_2,x_1) \ : \ x_2 = f(a), \quad x_1 \in (a,b) \ \} \text{ and} \\
\overline{S}_5 &= \{(x_2,x_1) \ : \ x_2 \in (f(b),f(a)), \quad x_1 \in (f^{-1}(x_2),b) \ \}.
\end{aligned}$$

Continuing along this line of logic identifies 23 distinct cases to consider (which may be reduced to effectively 12 after considering symmetry) for sets where $f \neq g$.

3.3.5 Region Swell

In general, after splitting the original S to ensure f and g are monotonic and invertible, pivoting a set will result in one to five disjoint subsets. This phenomena can be likened to intermediate coefficient swell in many polynomial arithmetic algorithms, and causes the number of regions necessary to represent a given function to increase while performing pivot and partial conjugate calculations.

After a pivot operation it is usually possible to simplify and merge adjacent regions into one region in an attempt to mitigate region swell. Such an operation helps to reduce intermediate region swell, and in most cases produces a pivoted function whose overall region complexity is comparable to the original.

3.3.6 Boundary Point Problem

While partial conjugates are always convex and lower semi-continuous with respect to the last variable conjugated, it is possible that the intermediates may not be lower semi-continuous with respect to the whole space. Referring to Example 3.2 we see that the first partial conjugate $f^{x_2}(x_1,y_2)$ is defined as $\frac{x_1}{4y_2}$ for $x_1, y_2 < 0$. Since $f^{x_2}(4mz,z) = m$, it follows that the limit of f^{x_2} as z approaches zero from the left (the limit in the direction

$-[4m, 1]$) is m. Thus, for any value $m \geq 0$, there is a sequence of points approaching this value at the origin. Hence, f^{x_2} is discontinuous at the origin.

In order for partial conjugation to succeed the input to the one dimensional partial conjugation operation must be lower semi-continuous. Thus, these points of discontinuity must be adjusted to be lower semi-continuous with respect to the next partial conjugate variable.

Consider the function f in recursive representation. For our purposes an admissible boundary point of f is any point $(a_{i_1}, a_{i_1,i_2}, \ldots, a_{i_1,\ldots,i_n})$ in the recursive representation of f that occurs on the boundary of dom f. Letting $x = (a_{i_1}, a_{i_1,i_2}, \ldots, a_{i_1,\ldots,i_n}) = (x_1, \ldots, x_n)$, x is a boundary point of f if and only if for all i one or both of $f(x_1, \ldots, x_i - \delta, \ldots, x_n) = \infty$ and $f(x_1, \ldots, x_i + \delta, \ldots, x_n) = \infty$ for sufficiently small $\delta > 0$ applies, but where for at least one i, only one applies.

To correct the boundary point problem we need only identify any points in the recursive representation that are boundary points and replace them with their limit as taken from the next partial conjugate variable, from the direction of the defined side, if there is one. Let $(x_1, \ldots, x_n)$ be such a point and consider the function values $g^-(\delta) = f(x_1, \ldots, x_n - \delta)$ and $g^+(\delta) = f(x_1, \ldots, x_n + \delta)$ for sufficiently small $\delta > 0$. If $g^-(\delta) = g^+(\delta) = \infty$, then we define $f(x_1, \ldots, x_n) = \infty$. If only $g^-(\delta) = \infty$, then we define $f(x_1, \ldots, x_n) = \lim_{\delta \downarrow 0} g^+(\delta)$. If neither of the former cases apply, then it follows by our definition of a boundary point that $g^+(\delta) = \infty$, in which case we define $f(x_1, \ldots, x_n) = \lim_{\delta \downarrow 0} g^-(\delta)$. This can be seen in the different values of f^{x_2} at the origin for its two distinct representations in Example 3.2. After conjugating with respect to x_2 we see that $f^{x_2}(0,0) = \lim_{y_2 \uparrow 0} 0 = 0$, whereas in the variable order (y_2, x_1) (where x_1 is the next partial conjugate variable) we see that we must set $f^{x_2}(0,0)$ to ∞ in order to preserve lower semi-continuity with respect to x_1.

3.4 Symbolic Conjugation in Higher Dimensions

With the ability to calculate conjugates of parameterized functions in $\mathcal{F}$ and the ability to pivot the representation of a function in $\mathcal{F}^n$, we can calculate an n dimensional conjugate as follows:

(1) Calculate the partial conjugate of $f(x_1, \ldots, x_n)$ resulting in $f^{x_n}(x_1, \ldots, x_{n-1}, y_n)$.

(2) Negate the partial conjugate.

(3) Pivot the partial conjugate to the variable order $(y_n, x_1, \ldots, x_{n-1})$.

(4) Calculate the partial conjugate of $f^{x_n}(y_n, x_1, \ldots, x_{n-1})$ resulting in $f^{(x_{n-1}, x_n)}(y_n, x_1, \ldots, x_{n-2}, y_{n-1})$.

(5) Repeat steps (2) through (4) for $x_{n-2}, \ldots, x_1$.

(6) Pivot the complete conjugate $f^*(y_2, \ldots, y_n, y_1)$ to the original variable order $f^*(y_1, \ldots, y_n)$.

3.5 Numerical Methods

We begin by noting that the convergence results of Section 2.5 all exist in generalized d dimensional forms (which can be found in [7] and [11]), thus discrete conjugation algorithms are equally valid and applicable in multiple dimensions.

Using the same iterated conjugation concepts as in the symbolic case, the one dimensional LLT from Section 2.5.1 may be easily extended to the d dimensional case. Consider a function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$, $X = X_1 \times \cdots \times X_d$, and $Y = Y_1 \times \cdots \times Y_d$. The algorithm works by calculating the partial one dimensional conjugates along X_1 yielding g^1 (defined over the grid $Y_1 \times X_2 \times \cdots \times X_d$). It then iteratively calculates g^i (defined over the grid $Y_1 \times \cdots \times Y_i \times X_{i+1} \times \cdots \times X_d$) as the partial conjugate along the i^{th} dimension of $-g^{i-1}$.

Let $n_i = |X_i|$ and $m_i = |Y_i|$. Calculating the partial conjugate along the i^{th} dimension involves calculating $\prod_{j<i} m_j \prod_{j>i} n_j$ conjugates, each an $O(n_i + m_i)$ operation. Summing this complexity over all i iterated partial conjugations yields a total complexity of $O(\sum_{i=0}^{d} n_1 \cdots n_i m_{i+1} \cdots m_d)$. Letting $n = n_1 \cdots n_d$ and $m = m_1 \cdots m_d$, this simplifies to the time complexity $O(d(n+m))$. For further details on algorithmic performance and implementation issues, refer to [11] and [12].

Chapter 4

Applications and Examples

In this chapter we aim to work through a representative set of examples displaying the use and capabilities of these algorithms in practice, as well as illustrating some potential applications.

In addition to simplifying and attempting to automate atomic convex analysis operations the tools may be used in more comprehensive practical settings as well. In some cases, it is possible to symbolically solve certain problems. In other cases, the tools can be used to aid symbolic solutions or inspire intuition through visualization.

4.1 Functionality of the `SCAT` Package

The algorithms of this thesis have been implemented in Maple as the Symbolic Convex Analysis Toolkit (`SCAT`). This package introduces significant new functionality and integrates itself into the Maple environment.

The following new commands have been introduced:

- `SCAT[Plot]`, for plotting one- and two dimensional convex functions and one dimensional subdifferentials;
- `SCAT[PwfToPiecewise,PiecewiseToPwf,PwfToRegions,RegionsToPwf]`, for converting between different representations of piecewise continuous functions;
- `SCAT[Eval]`, for evaluating any dimensional convex functions and one dimensional subdifferentials at points;

- `SCAT[Subs]`, for performing substitutions into any dimensional convex functions and one dimensional subdifferentials;
- `SCAT[Limit]`, for calculating limits of free parameters in any dimensional convex functions and one dimensional subdifferentials;
- `SCAT[SubDiff,Integ]`, for calculating the subdifferentials of one dimensional convex functions, and integrating them back to one dimensional convex functions;
- `SCAT[Conj,PartialConj]`, for calculating complete and partial conjugates of any dimensional convex functions;
- `SCAT[InfConv]`, for calculating the infimal convolution of a set of any dimensional convex functions;
- `SCAT[Convex]`, for attempting to prove convexity of a given any dimensional function;
- `SCAT[Equal]`, for comparing any dimensional functions and one dimensional subdifferentials;
- `SCAT[Assume,Additionally,Assumptions]`, for dealing with constraints and assumptions on free parameters within `SCAT` internal data formats for functions; and,
- `CCAT[CreateNpwf,ConjN,SubDiffN]`, for creating numeric `NPWF` function representations, and calculating numeric conjugates and subdifferentials.

Additionally, the `SCAT` package has been integrated as much as possible into Maple, supporting the following built-in functions:

- `type`, for type testing of `SCAT` internal data formats;
- `print`, for pretty-printing any dimensional convex functions and one dimensional subdifferentials;
- `convert`, for converting piecewise functions to the `SCAT` internal `PWF` format; and,
- `simplify`, for simplifying algebraic operations applied to one or more any dimensional convex functions and one dimensional subdifferentials.
- Standard Maple commands such as `norm`, `evalf`, `factor`, `expand`, etc, also work with `PWF` and `SD` objects.

For a complete listing of all commands and their usage, refer to Appendix A.

4.2 Ten Classic Examples

The following examples aim mainly to demonstrate the usage of the software. To this end, a representative set of one- and many dimensional examples have been selected from [2, 3, 4, 16]. Specific emphasis has been placed on demonstrating introduced or improved functionality not possible in earlier packages.

Example 4.1 (Absolute value) One of the simplest examples of a convex function that is not everywhere differentiable is the absolute value function $f : x \mapsto |x|$. Its derivative at the origin fails to exist since $f'_-(0) = -1 < 1 = f'_+(0)$. The notion of the subgradient is able to capture this behaviour and accordingly it is seen that $\partial f(0) = [-1, 1]$. In order to explore this function we first represent it in a form that SCAT understands; the PWF (piecewise function) format:

```
> f1 := convert(abs(x),PWF);
```

$$f1 := \begin{cases} -x, & x < 0 \\ 0, & x = 0 \\ x, & x > 0 \end{cases}$$

We may easily calculate the subdifferential of f1 and confirm our earlier calculation from Example 1.19:

```
> sdf1 := SubDiff(f1);
```

$$sdf1 := \begin{cases} \{-1\}, & x < 0 \\ [-1, 1], & x = 0 \\ \{1\}, & x > 0 \end{cases}$$

We may also calculate the conjugate, yielding the expected answer as found in Example 1.23:

```
> g1 := Conj(f1,y);
```

$$g1 := \begin{cases} \infty, & y < -1 \\ 0, & y = -1 \\ 0, & (-1 < y) \textbf{ and } (y < 1) \\ 0, & y = 1 \\ \infty, & 1 < y \end{cases}$$ □

Example 4.2 (Negative entropy) The exponential function and the (negative) Boltzmann-Shannon entropy function are a well known pair of Fenchel conjugates. Using the SCAT package this conjugacy relationship is easily confirmed by entering:

```
> f2 := convert(exp(x),PWF);
```

$$f2 := \{ e^x \qquad \text{all}(x)$$

```
> g2 := Conj(f2,y);
```

$$g2 := \begin{cases} \infty, & y < 0 \\ 0, & y = 0 \\ \ln(y)y - y, & 0 < y \end{cases}$$

□

Example 4.3 (De Pierro and Iusem)

This function was originally suggested by De Pierro and Iusem on page 438 of [8], and also used as an example in [2, 3]. The function is easily constructed and its conjugate calculated by issuing the following commands:

```
> piecewise(x<=1,1/2*(x^2-4*x+3),-ln(x)):
> f3 := convert(%,PWF);
```

$$f3 := \begin{cases} \frac{1}{2}x^2 - 2x + \frac{3}{2}, & x < 1 \\ 0, & x = 1 \\ -\ln(x), & 1 < x \end{cases}$$

```
> g3 := Conj(f3,y);
```

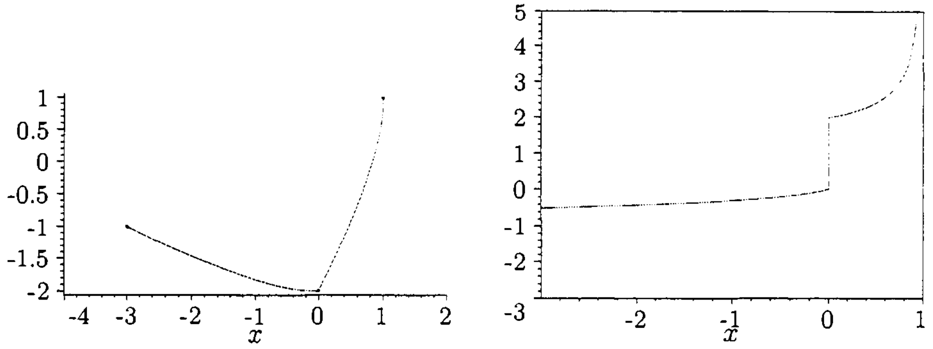
$$g3 := \begin{cases} 2y + \frac{1}{2}y^2 + \frac{1}{2}, & y < -1 \\ -1, & y = -1 \\ -1 - \ln(-y), & (-1 < y) \textbf{ and } (y < 0) \\ \infty, & y = 0 \\ \infty, & 0 < y \end{cases}$$

□

Example 4.4 (Affine and quadratic) Affine functions on $\mathbb{R}$ are those of the form $f : x \mapsto bx + c$ where b and c are both real constants. Being a function of a constant slope, there is only one subgradient (that of slope b) that minorizes it. Thus, the conjugates of these functions are finite at only one point, as shown by entering the command `Conj(convert(b*x+c,PWF,x))`:

$$\begin{cases} \infty, & y < b \\ -c, & y = b \\ \infty, & b < y \end{cases}$$

Similarly, quadratic functions are those of the form $f : x \mapsto ax^2 + bx + c$ for $a \neq 0$, and real constants b and c. The subset of convex quadratic functions (those with $a > 0$) turns out to be closed under the operation of Fenchel conjugation, as shown with the commands `f4 := convert(a*x^2+b*x+c,PWF,x,{a>0})` and `g4 := Conj(f4,y)`:



(a) Plot of f5

(b) Plot of sdf5

Figure 4.1: Plots from Example 4.5

$$g4 := \left\{ -\frac{1}{4}\frac{2yb-y^2-b^2+4ca}{a} \qquad \text{all}(y) \right.$$

A specific case of the more general result from Theorem 1.25 may be seen by solving for the values of a, b and c that make the above pair of conjugates equal. We can quickly generate a system of three equations and solve for the parameters with the following commands:

```
> F4 := PwfToPiecewise(f4):
  G4 := subs(y=x,PwfToPiecewise(g4)):
  map(i->subs(x=i,F4=G4),[0,1,-1]):
  solve({op(%),a>0},{a,b,c});
  subs(op(%),F4=G4);
```

$$\{c = 0, a = \tfrac{1}{2}, b = 0\}$$
$$\tfrac{1}{2}x^2 = \tfrac{1}{2}x^2$$

□

Example 4.5 (An example from Rockafeller) The following function can be found on page 229 of Rockafeller's text [16]. The function is easily constructed using piecewise and converted to the PWF format:

```
> piecewise(-3<=x and x<=1,abs(x)-2*sqrt(1-x),infinity):
  f5 := convert(%,PWF);
```

$$f5 := \begin{cases} \infty, & x < -3 \\ 1, & x = -3 \\ -2\sqrt{1-x} - x, & (-3 < x) \textbf{ and } (x < 0) \\ -2, & x = 0 \\ -2\sqrt{1-x} + x, & (0 < x) \textbf{ and } (x < 1) \\ 1, & x = 1 \\ \infty, & 1 < x \end{cases}$$

We now use the command `Plot(f5,x=-4..2,scaling=constrained,axes=framed)` to plot the function, yielding Figure 4.1(a). Next, to calculate and plot the subdifferential we use the commands `sdf5 := SubDiff(f5)`, and

```
Plot(sdf5,-3..1,view=[-3..1,-3..5],axes=none),
```

yielding

$$sdf5 := \begin{cases} \{\}, & x < -3 \\ [-\infty, -\frac{1}{2}], & x = -3 \\ \{\frac{(-1+\sqrt{1-x})\sqrt{1-x}}{x-1}\}, & (-3 < x) \textbf{ and } (x < 0) \\ [0, 2], & x = 0 \\ \{-\frac{(1+\sqrt{1-x})\sqrt{1-x}}{x-1}\}, & (0 < x) \textbf{ and } (x < 1) \\ \{\}, & x = 1 \\ \{\}, & 1 < x \end{cases}$$

and the plot in Figure 4.1(b). Finally, we find the conjugate, the biconjugate and manually verify the convexity of `f5` with the following commands:

```
> g5 := Conj(f5,y);
```

$$g5 := \begin{cases} -3y + 1, & y < -\frac{1}{2} \\ \frac{5}{2}, & y = \frac{-1}{2} \\ \frac{y^2+2y+2}{1+y}, & (\frac{-1}{2} < y) \text{ and } (y < 0) \\ 2, & y = 0 \\ 2, & (0 < y) \text{ and } (y < 2) \\ 2, & y = 2 \\ \frac{y^2-2y+2}{-1+y}, & 2 < y \end{cases}$$

```
> F5 := Conj(g5,x):
  Equal(f5,F5);
```

$$\mathit{true}$$

□

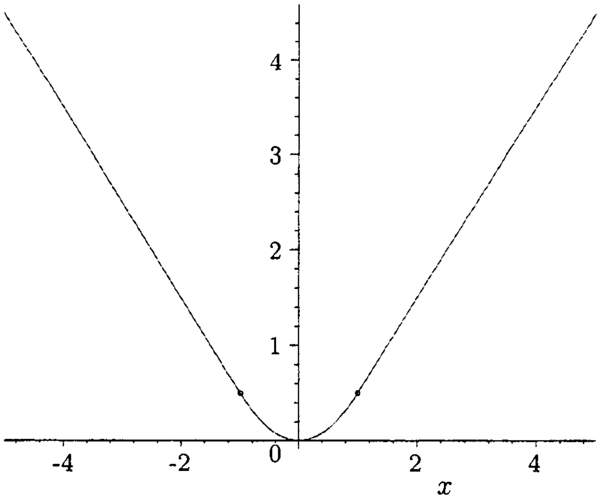


Figure 4.2: Plot of (f1* + f6*)* from Example 4.6

Example 4.6 (An infimal convolution) Given two closed convex functions f and g the function $(f^* + g^*)^*$ is called the (closure of the) *infimal convolution* of f and g. If either one of the functions is differentiable then the infimal convolution will be as well; thus, the operation is a *regularization*, which can be used to add additional structure to an object while maintaining much of its original shape. In this example we regularize the non-differentiable absolute value function from Example 4.1 with $\frac{1}{2}x^2$. A plot of the regularized function can be found in Figure 4.2. Notice that it retains the large-scale features of the absolute value function, but with the discontinuity smoothed out by the quadratic.

```
> f6 := convert(x^2/2,PWF):
> Conj(simplify(Conj(f1,y)+Conj(f6,y)),x);
> Plot(%,-5..5);
```

$$\begin{cases} -x - \frac{1}{2}, & x < -1 \\ \frac{1}{2}, & x = -1 \\ \frac{1}{2}x^2, & (-1 < x) \textbf{ and } (x < 1) \\ \frac{1}{2}, & x = 1 \\ x - \frac{1}{2}, & 1 < x \end{cases}$$

We can also perform the infimal convolution by calling the command `InfConv(f1,f6)` directly. □

Example 4.7 (Young's inequality) Suppose $1 < p < \infty$ and let q be such that $\frac{1}{p}+\frac{1}{q} = 1$. The equality

$$\frac{1}{p}a^p + \frac{1}{q}b^q \geq ab, \quad \forall a, b \geq 0,$$

is known as *Young's inequality*. As we are about to see, since $(\frac{1}{p}|\cdot|^p)^* = \frac{1}{q}|\cdot|^q$ this is actually a special case of the stronger Fenchel-Young inequality from Theorem 1.24. In this example we show and confirm the above conjugate pair.

This example elaborates on a similar example provided in [3]. The algorithms developed in this thesis are able to handle p as a free parameter while those in [3] were restricted to fixed values of p. The general pair of conjugate functions is easily derived using the following commands:

```
> f7 := convert(abs(x)^p/p,PWF,x,{p>1});
  g7 := Conj(f7,y):
  g7 := Subs(p=1/(1-1/q),g7);
```

$$f7 := \begin{cases} \frac{(-x)^p}{p}, & x < 0 \\ 0, & x = 0 \\ \frac{x^p}{p}, & 0 < x \end{cases}$$

$$g7 := \begin{cases} \frac{(-\frac{1}{y})^{(-q)}}{q}, & y < 0 \\ 0, & y = 0 \\ \frac{y^q}{q}, & 0 < y \end{cases}$$

In creating f7, notice that we passed additional parameters consisting of a set of assumptions. In this example, if we do not provide the information that $p > 1$ then the process will fail, producing the following output:

```
> f := convert( abs(x)^p/p, PWF, x );
Error, (in EvalRel) unable to evaluate relation:
1/p*limit(x^p,x = 0,right) = 1/p*limit((-x)^p,x = 0,left)
```

□

Example 4.8 (Indicator function of the unit ball in $\mathbb{R}^2$) We now consider the indicator function of the unit ball in two-dimensions. Due to the verbose nature of the output for multi dimensional PWF objects, we will generally suppress the display of these objects.

We begin by manually constructing the PWF object, which is recursive in nature, and described in section 3.3.2. The indicator function of the unit ball is simply the function that has value 0 for all $|x| \leq 1$, and ∞ elsewhere. The PWF object for this function is constructed with the following commands:

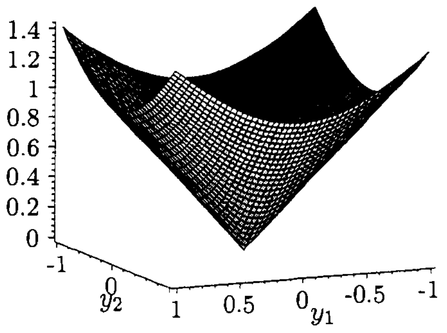


Figure 4.3: Plot of g8 from Example 4.8

```
> [infinity]:
> [infinity,0,0,infinity]:
> [infinity,sqrt(1-x1^2),0,0,sqrt(1-x1^2),0,infinity]:
> [%%%,-1,%%,%,1,%%,%%%]:
> f8 := PWF(%,[x1,x2],x1::real,x2::real):
```

The conjugate of this function is calculated using the command `g8 := Conj(f8,[y1,y2])`, which yields the function $\sqrt{y_1^2+y_2^2}$. It turns out that this and Example 4.1 are simply specific cases of the more general result on $\mathbb{R}^n$ that

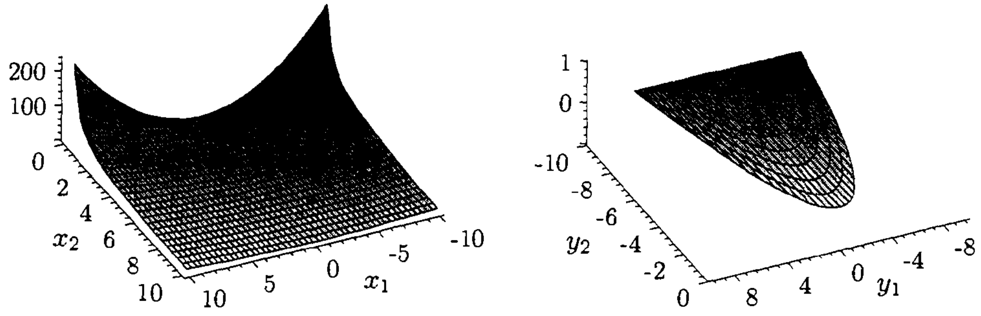
$$\|x\|^* = \begin{cases} 0, & \|y\| \leq 1 \\ \infty, & \text{otherwise.} \end{cases}$$

The plot of g8 in Figure 4.3 is generated using the command `Plot(g8, -1..1, -1..1, axes=framed, orientation=[66,77])`. □

Example 4.9 (An example on $\mathbb{R}^2$ from Borwein and Lewis) We consider the following function given in an exercise on page 40 of [4]:

$$\begin{cases} \frac{x_2^2}{x_1}, & \text{if } x_2 > 0 \\ 0, & \text{if } \mathrm{x} = 0 \\ \infty, & \text{otherwise.} \end{cases}$$

Specifically, we consider a variation of the above function defined on the half-plane for $x_2 > a > 0$, and consider the behaviour of this function in the limit as a decreases to 0.



(a) Plot of f9b

(b) Plot of g9b = f9b*

Figure 4.4: Conjugate pair from Example 4.9

The following code manually creates the PWF object corresponding to this function:

```
> [infinity,a,0,0]:
> [infinity,a,x1^2/a,x1^2/x2]:
> [%,0,%%,%]:
> f9 := PWF(%,[x1,x2],{x1::real,x2::real,a>0}):
```

We can learn about the behaviour of this function at its limit through the following commands:

```
> g9a := Limit(Conj(f9,[y1,y2]),a=0,right):
> g9b := Conj(Limit(f9,a=0,right),[y1,y2]):
> Equal(g9a,g9b);
                              true
```

Thus we see that in this example the conjugate of the limit and the limit of the conjugate agree. Finally, we can prove convexity (answering the exercise presented in [4]) and visually examine the conjugate pair (Figure 4.4) with the following commands:

```
> f9a := Limit(f9,a=0,right):
> Convex(f9);
> Plot(f9a,-10..10,0..10,axes=framed,orientation=[65,30]);
> Plot(g9a,-10..10,-10..0,axes=framed,orientation=[65,30]);
                              true
```

□

Example 4.10 (An example on $\mathbb{R}^3$) We consider one final example in higher dimensions. In this example we demonstrate an alternative construction technique, building the PWF object from its (non-recursive) region representation, as discussed in section 3.3.1. It is often the case that such a representation is easier and more readable for PWF creation purposes. We consider the function $-\ln(x+1)+y\ln(y)+z^2$ defined on $\mathbb{R}^3_+$. The following commands generate the PWF and test it for convexity.

```
> f := (x,y,z) -> -log(x+1)+y*log(y)+z^2:
> R := [[0,infinity, 0,infinity, 0,infinity, f(x,y,z)],
  [0,infinity, 0,infinity, 0,0, f(x,y,0)],
  [0,infinity, 0,0, 0,infinity, f(x,0,z)],
  [0,0, 0,infinity, 0,infinity, f(0,y,z)],
  [0,infinity, 0,0, 0,0, f(x,0,0)],
  [0,0, 0,infinity, 0,0, f(0,y,0)],
  [0,0, 0,0, 0,infinity, f(0,0,z)],
  [0,0, 0,0, 0,0, f(0,0,0)]]:
  vl := [x,y,z]:
  al := convert(map(i->i::real,vl),set):
  RegionsToPwfPl(R,vl):
  f10 := PWF(%,vl,al):
  Convex(f10);
```

$$\mathit{true}$$

SCAT is able to extract lower dimensional convex functions from higher dimensional functions through partial evaluation. We demonstrate this ability in the following example:

```
> g := Conj(f,[X,Y,Z]):
  Eval(g,Y=-1,Z=0);
```

$$\begin{cases} e^{(-2)}, & X < -1 \\ e^{(-2)}, & X = -1 \\ -1 - X - \ln(-X) + e^{(-2)}, & (-1 < X) \text{ and } (X < 0) \\ \infty, & X = 0 \\ \infty, & 0 < X \end{cases}$$

□

4.3 Horse Racing Problem

SCAT is powerful enough to handle many functions symbolically, and this can allow for certain optimization problems to be solved symbolically. In the following example from [13], SCAT is able to find a closed form of the dual which allows us to find quick and accurate numerical solutions to the primal.

Suppose there is a fixed quantity of x_0 of some commodity that needs to be allocated among n distinct activities in such a way as to maximize the return. We may assume that the return associated with the i^{th} activity is an increasing concave function $g_i(x)$ due to diminishing marginal returns. Letting x_i represent the amount of commodity allocated to the i^{th} activity, the problem may be stated as

$$\begin{cases} \text{maximize } g(x) = \sum_{i=1}^{n} g_i(x_i) \\ \text{subject to } \sum_{i=1}^{n} x_i = x_0, \quad \text{and} \quad x_i \geq 0,\ i = 1, \ldots, n. \end{cases}$$

This problem is easily recast into the framework of Fenchel duality. Let each g_i have domain $\mathbb{R}_+$, and accordingly let g have the domain $\mathbb{R}^n_+$. Define the set $C = \{x : \sum_{i=1}^{n} x_i = x_0\}$ and construct the function $f = \delta_C$. Since each g_i is concave on $\mathbb{R}$, then g is concave on $\mathbb{R}^n$. Since C is a convex set it follows that f is convex. We see that $\operatorname{dom} f \cap \operatorname{dom} g = \{x : \sum_{i=1}^{n} x_i = x_0,\ x \in \mathbb{R}^n_+\}$, thus our problem is defined over the correct domain. In this notation, the problem now becomes

$$\inf\{-g(x) : x \in C \cap \mathbb{R}^n_+\}.$$

We now consider the convex conjugate f^*, given by

$$\begin{aligned} f^*(y) &= \sup\{\langle y, x\rangle - f(x) : x \in C\} \\ &= \sup\{\langle y, x\rangle : x \in C\}. \end{aligned}$$

Let a be the index of the y_i with the largest magnitude, and similarly let b be the index of the y_i with the smallest magnitude. Suppose $|y_b| < |y_a|$. By setting $x_a = x_0 + r\operatorname{sign}(y_a)$, $x_b = -r\operatorname{sign}(y_a)$ and $x_i = 0$ otherwise, we see that as r tends to infinity, so does $\langle y, x\rangle$. Now consider y such that $|y_a| = |y_b|$. There are two subcases to consider. Suppose $y_a = -y_b$. Without loss of generality, let $y_a > 0$. Taking the same allocation as above yields $\langle y, x\rangle = (x_0 + r)y_a - r(-y_a) = x_0 + 2ry_a$, which tends to infinity as r does. Thus, we are left with the case $y_a = y_b$, which implies that y has the form $y = \lambda(1, \ldots, 1)$ for $\lambda \in \mathbb{R}$. In this case, the inner product always has the same value, namely λx_0. Thus, we see that

$$f^*(y) = \begin{cases} \lambda x_0, & y = \lambda(1, \ldots, 1) \\ \infty, & \text{otherwise.} \end{cases}$$

Since g is separable, we may easily calculate $g_*(y)$ as

$$g_*(y) = \sum_{i=1}^{n} \{(g_i)_*(y)\}.$$

The dual problem then becomes

$$\inf_{\lambda \in \mathbb{R}} \{\lambda x_0 - \sum_{i=1}^{n} \{(g_i)_*(\lambda)\}. \tag{4.11}$$

Surprisingly, the n dimensional primal problem is reduced to a single dimension optimization problem in the dual.

Consider the problem of betting on a horse race. Assuming we know the probability p_i that the i^{th} horse will win, we wish to know how best to distribute a total bet of x_0 dollars. Let the track keep a proportion $0 < 1 - C < 1$ of the total amount bet and distribute the rest proportionally amongst those who bet on the winning horse. Finally, let s_i be the amount that the rest of the public is betting on horse i. If we bet amount x_i on the i^{th} horse, we receive

$$C\left(x_0 + \sum_{i=1}^{n} s_i\right) \frac{x_i}{s_i + x_i}$$

if it wins. Thus, the expected net return R is calculated as

$$R = C\left(x_0 + \sum_{i=1}^{n} s_i\right)\left(\sum_{i=1}^{n} \frac{p_i x_i}{s_i + x_i}\right) - x_0.$$

The problem then becomes to maximize R, or equivalently

$$g(x) = \sum_{i=1}^{n} g_i(x_i)$$

where

$$g_i(x) = \frac{p_i x_i}{s_i + x_i}.$$

Inspecting the second derivative of g_i shows that it is strictly decreasing, and thus each g_i is concave. Using the relationship between convex and concave conjugates in Section 1.6.1, SCAT can calculate the concave conjugate of g_i yielding:

$$\begin{cases} -\infty, & \lambda < 0 \\ -p_i, & \lambda = 0 \\ -\lambda s_i + 2\sqrt{\lambda p_i s_i} - p_i, & (0 < \lambda) \text{ and } (\lambda < \frac{p_i}{s_i}) \\ 0, & \lambda = \frac{p_i}{s_i} \\ 0, & \frac{p_i}{s_i} < \lambda \end{cases}$$

The calculated closed form of each g_i may then be used in solving the one dimensional minimization in Equation 4.11. Unfortunately, no symbolic solution exists to this minimization, but any numerical solver will quickly and accurately find the unique minimization point because of the symbolic representation of the objective function.

Given a solution λ to the dual problem, we want to find the associated x_i values in the primal domain. By Theorem 1.26, it follows that the optimal x will maximize the equation

$$\langle x, \lambda(1, \ldots, 1)\rangle - g(x).$$

Thus, each x_i will maximize

$$x_i\lambda - g_i(x_i).$$

Solving for the critical point by differentiation yields

$$x_i = \sqrt{\frac{s_i p_i}{\lambda}} - s_i.$$

Since this value is negative (and outside of the domain of g_i) if $\lambda > \frac{p_i}{s_i}$, then it follows that

$$x_i = \begin{cases} \sqrt{\frac{s_i p_i}{\lambda}} - s_i, & \lambda < \frac{p_i}{s_i} \\ 0, & \text{otherwise.} \end{cases}$$

Thus, we see that λ is chosen such that

$$S(\lambda) = \sum_{i:\lambda<\frac{p_i}{s_i}} \left(\sqrt{\frac{s_i p_i}{\lambda}} - s_i\right) = x_0.$$

Now $S(\lambda)$ is easily verified as continuous, and it can be seen that $S(0) = \infty$, $S(\infty) = 0$. Thus, there will always exist a λ that satisfies this equation.

It is interesting to note that for small x_0, a larger λ will need to be found to satisfy this equation, and as x_0 gets small enough ($x_0 << \sum s_i$), it will eventually be such that λ is smaller than only the maximum $\frac{p_i}{s_i}$. This means that the entire bet should be placed on the single horse with maximum $\frac{p_i}{s_i}$, or equivalently, with maximum $p_i r_i$ where

$$r_i = C \sum_j \frac{s_j}{s_i}$$

is the track odds.

4.4 Future Work

While progress has been made in extending earlier work on symbolic conjugation to the non-separable multi dimensional case, much work remains to be done. The two biggest hurdles to successfully completing a conjugation calculation are the inverting of the one dimensional subdifferential, and the pivoting operation between partial conjugates.

Focussing effort on improving the ability to find inverses on a wider variety of functions would simultaneously improve the functionality of SCAT on both of these troublesome fronts. This is most directly addressed by improving the underlying tools in Maple.

The operation of pivoting (variable reordering) in two-dimensions has been fully explored, and is limited only by the ability to find branch points and inverses. However, there remains much room for further exploration into variable reordering in higher dimensions. While there is not much hope for a general solution, many special cases and heuristics are sure to exist which will extend the class of functions SCAT can handle in closed form.

There is also the possibility of tackling new related problems. Having the ability to symbolically calculate convex hulls of one dimensional functions would greatly improve the range of input functions that SCAT could handle, as well as provide useful new functionality in its own right. Additionally, it would be interesting to investigate direct algorithms for calculating infimal convolutions symbolically instead of using conjugation and addition; a direct algorithm would likely be more efficient, and may be able to handle a broader class of input functions.

Other often neglected areas to improve are those of user interface and data structures. It is currently rather cumbersome to create symbolic representations of higher dimensional functions, as evidenced by the latter examples in Section 4.2. Improved data structures may simultaneously yield more intuitive representations and allow for algorithmic improvements.

This thesis has presented algorithms for symbolically calculating Fenchel conjugates on $\mathbb{R}^n$ and subdifferentials on the real line. It has provided examples of situations where the algorithms succeed, commented on their shortcoming and identified areas for improvement. It is hoped that the SCAT package will be a useful tool that will spur further research into both symbolic and numeric algorithms for problems in convex analysis.

Appendix A

Command Listing

This appendix serves as a glossary of all commands introduced in the Convex Analysis Toolkit for Maple. Each of the 27 new commands is described with comments on usage, input and output. The latest version of the toolkit may be found on the internet in the *Dalhousie D-DRIVE Document Server*, along with commented source code and example Maple work-sheets.

A.1 Symbolic Convex Analysis Toolkit

A.1.1 Introduction

The Symbolic Convex Analysis Toolkit was written in Maple 7, but has been tested with all versions of Maple up to and including version 9.5. Where there are known issues, they have been noted along with the command description.

A.1.2 Commands

A.1.2.1 convert/PWF

Description: Integrates with Maple `convert` command to convert functions, expressions and piecewise objects to the `PWF` format.

Usage:

```
convert( f, PWF, v, al )
```

Input:

`f` A function, expression or piecewise object.

`v` A list of variables and their ranges, as with the `plot` command. If no range specified, the function is assumed to be defined on the whole real line. If a range is specified the function is defined as infinity outside the range.

`al` An (optional) set of assumptions on other parameters appearing in the expression.

Output: A `PWF` object representing `f`.

A.1.2.2 SCAT[PwfToPiecewise]

Description: Converts a one dimensional `PWF` object to the equivalent piecewise object.

Usage:

```
PwfToPiecewise( pwf )
```

Input:

`pwf` A one dimensional `PWF` object.

Output: An equivalent `piecewise` object.

A.1.2.3 SCAT[PwfToRegions]

Description: Converts a `PWF` object to a list of regions.

Usage:

```
PwfToRegions( pwf )
```

Input:

`pwf` A `PWF` object.

Output: A list of regions.

A.1.2.4 SCAT[RegionsToPwf]

Description: Converts a list of regions to a PWF object.

Usage:

```
RegionsToPwf( R, X, A )
```

Input:

R	A list of n dimensional regions.
X	A list of n variables.
A	An (optional) set of assumptions on parameters appearing in the list R.

Output: An equivalent PWF object.

A.1.2.5 SCAT[SubDiff]

Description: Calculates the subdifferential of a one dimensional PWF object.

Usage:

```
SubDiff( pwf )
```

Input:

pwf	A one dimensional PWF object.

Output: The subdifferential as an SD object.

A.1.2.6 SCAT[Integ]

Description: Integrates a given SD object, and returns a corresponding one dimensional PWF object. Arbitrarily choose the constant of integration.

Usage:

```
Integ( sd )
```

Input:

sd	An SD object.

Output: A PWF object.

A.1.2.7 SCAT[Plot]

Description: Plots one- or two dimensional PWF objects, and SD objects.

Usage:

```
Plot( pwf1, r, opt2, ...  )
Plot( pwf2, r1, r2, opt3, ...  )
Plot( sd, r, opt2, ...  )
```

Input:

pwf1	A one dimensional PWF object.
pwf2	A two dimensional PWF object.
sd	An SD object.
r, r1, r2	Ranges of the form a..b or x=a..b.
opt2	Any (optional) plot compatible options.
opt3	Any (optional) plot3d compatible options.

Output: A corresponding plot.

A.1.2.8 SCAT[Assume]

Description: Erases all previous assumptions associated with a given PWF or SD object, and stores any new ones provided.

Usage:

```
Assume( pwf, a, ...  )
Assume( sd, a, ...  )
```

Input:

pwf	A PWF object.
sd	An SD object.
a	An (optional) assumption.

Output: A new object containing the corresponding assumptions.

A.1.2.9 SCAT[Additionally]

Description: Adds additional assumptions to an existing set associated with a given PWF or SD object.

Usage:

```
Additionally( pwf, a, ... )
Additionally( sd, a, ... )
```

Input:

pwf	A PWF object.
sd	An SD object.
a	An (optional) assumption.

Output: A new object containing the corresponding assumptions.

A.1.2.10 SCAT[Additionally]

Description: Returns the set of assumptions associated with a given PWF or SD object.

Usage:

```
Assumptions( pwf )
Assumptions( sd )
```

Input:

pwf	A PWF object.
sd	An SD object.

Output: The set of assumptions.

A.1.2.11 SCAT[Eval]

Description: Evaluates a PWF or SD object at a given location. For higher dimensional objects, some portion of the variables may be specified, resulting in a corresponding lower dimensional version of the object.

Usage:

```
Eval( pwf, val1, val2, ...  )
Eval( sd, val )
```

Input:

pwf	A PWF object.
sd	An SD object.
val, val1, val2	A variable value. The nth value is assumed to refer to the nth variable, unless directly specified in the form x=a.

Output: The value of the object at the given location, or possibly a lower dimensional partially evaluated PWF object.

A.1.2.12 SCAT[Equal]

Description: Tests two PWF or SD objects for equality. They must be in the same variable order. Multi dimensional objects in a different variable order may in fact be equal, but will be reported as not equal.

Usage:

```
Equal( pwf1, pwf2 )
Equal( sd1, sd2 )
```

Input:

pwf1, pwf2	A PWF object.
sd1, sd2	An SD object.

Output: A boolean value.

A.1.2.13 SCAT[Limit]

Description: Applies a limit to a PWF or SD objects. Due to the object structures, normal limit operations will not work, and this procedure must be used instead. Limits may only be taken with respect to free parameters (and not variables of the object).

Usage:

```
Limit( pwf, x=val, direction )
Limit( sd, x=val, direction )
```

Input:

`pwf`	A `PWF` object.
`sd`	An `SD` object.
`x`	A variable.
`val`	A value.
`direction`	An (optional) direction *from* which the limit is taken. The choices are `left`, `right`, `real`. If not provided, assumed to be `real`. See `?limit` for more details.

Output: The object with the appropriate limit taken.

A.1.2.14 SCAT[Subs]

Description: Performs substitutions on `PWF` or `SD` objects. Using a regular call to `subs` may break the underlying structure, so this is a safe wrapper to avoid any problems. The syntax is identical to the existing `subs` routine.

NOTE: This can't currently properly handle variable substitutions. For example, substituting $x = -x$ will result in a malformed `PWF` object.

Usage:

```
Subs( x1=a1, ..., pwf )
Subs( x1=a1, ..., sd )
```

Input:

`pwf`	A `PWF` object.
`sd`	An `SD` object.
`x1`	An expression to be replaced.
`a1`	An expression to be substituted.

Output: The modified object.

A.1.2.15 SCAT[PartialConj]

Description: Performs the partial conjugation of a given `PWF` with respect to the variable that is last in the variable list. Not guaranteed to succeed.

NOTE: Success of the partial conjugate depends on Maple's ability to invert functions and find zeroes of derivatives. Because of the difficulty of this task, the algorithm will not always succeed.

Usage:

```
PartialConj( pwf, y )
```

Input:

`pwf`	A `PWF` object.
`y`	The conjugate variable name.

Output: The partial conjugate.

A.1.2.16 SCAT[Pivot]

Description: Pivots a given `PWF` object. That is, reorganizes the function from the variable order $[x_1, \ldots, x_n]$ to $[x_n, x_1, \ldots, x_{n-1}]$.

NOTE: This operation depends on the ability to find inverses, roots of derivatives, etc. This operation is not guaranteed to succeed, especially for complex examples in dimensions higher than 2.

Usage:

```
Pivot( pwf )
```

Input:

`pwf`	A `PWF` object.

Output: The pivoted object.

A.1.2.17 SCAT[Conj]

Description: Calculates the Fenchel conjugate of a given `PWF`.

NOTE: This operation depends on the `PartialConj` and `Pivot` operations, and thus may also fail, especially for complicated examples in dimensions higher than 2.

Usage:

```
Conj( pwf, vl )
```

Input:

`pwf`	A `PWF` object.
`vl`	A list of conjugate variable names.

Output: The conjugated function.

A.1.2.18 SCAT[Convex]

Description: Attempts to determine if a given function is convex or not. Currently does this by calculating the bi-conjugate, and testing for equality.

NOTE: May return *FAIL* if the underlying operations are unable to finish.

Usage:

```
Convex( pwf )
```

Input:

`pwf`	A `PWF` object.

Output: A boolean value upon successful completion, *FAIL* otherwise.

A.1.2.19 SCAT[InfConv]

Description: Attempts to perform the infimal convolution of one or more functions.

NOTE: This operation depends on `Conj` and may therefore not always succeed.

Usage:

```
InfConv( pwf1, pwf2, ...  )
```

Input:

`pwf1, pwf2`	A `PWF` object.

Output: The infimal convolution of the list of functions, as a `PWF` object.

A.2 Computational Convex Analysis Toolkit

A.2.1 Introduction

The Computational Convex Analysis Toolkit was written in Maple 7, but has been tested with all versions of Maple up to and including version 9.5.

While the algorithms implemented in this toolkit are naturally extended to arbitrary dimensions, running in an interpreted language places heavy heavy overhead on their implementation. Additionally, these numeric algorithms are intended as an aid to visualization, which is fundamentally limited to functions defined over $\mathbb{R}^2$. For this reason, all of the algorithms in this package will currently only work for dimensions 1 and 2.

A.2.2 Commands

A.2.2.1 CCAT[CreateN]

Description: Creates an `NPWF` object.

Usage:

```
CreateN( f, [rx1, ..., rxn], m )
```

Input:

`f` An n dimensional procedure accepting values of type *finite*, and outputting values in $(-\infty, \infty]$.

`rx1, ..., rxn` An array/list of finite values, or a *range* of *extended_numeric* values.

`m` An (optional) number of points to use per dimension. Over-ridden if `rxi` is an array or list of points.

Output: The `NPWF` object representing `f` over the given domain.

A.2.2.2 CCAT[PlotN]

Description: Plots a given one- or two dimensional `NPWF` object.

Usage:

```
PlotN( npwf, opt )
```

Input:

`npwf`	An `NPWF` object.
`opt`	Any (optional) plot options (see `?plot,options`).

Output: A two or three dimensional plot.

A.2.2.3 CCAT[EvalN]

Description: Evaluates an n dimensional `NPWF` object at a given point in $\mathbb{R}^n$. By convention, the function is defined to be ∞ outside of the domain specified when created. Any points off the grid of definition are calculated as the appropriate convex combination of their neighbors (linear interpolation).

Usage:

```
EvalN( npwf, X )
```

Input:

`npwf`	An n dimensional `NPWF` object.
`X`	A list of n *finite* values.

Output: The value of `npwf` at `X`.

A.2.2.4 CCAT[GetDualRangeN]

Description: Given an `NPWF` object, returns the interesting ranges for each dual variable. To be used as an aid to help determine interesting regions over which to calculate the conjugate. Essentially, the minimum and maximum slopes of the convex hull along each dimension.

Usage:

```
GetDualRangeN( npwf )
```

Input:

`npwf`	An n dimensional `NPWF` object.

Output: An n dimensional list of ranges.

A.2.2.5 CCAT[ConvexHullN]

Description: Calculates the convex hull of the given NPWF object.

Usage:

```
ConvexHullN( npwf )
```

Input:

`npwf` An NPWF object.

Output: The convex hull, as an NPWF object.

A.2.2.6 CCAT[ConjN]

Description: Calculate the conjugate of a given NPWF object.

Usage:

```
ConjN( npwf, [rx1,...,rx2], m )
```

Input:

`npwf` An NPWF object.

`rx1, ..., rxn` An (optional) array/list of finite values, or a range of *extended_numeric* values. May be completely or partially omitted, and individual ranges may be specified with DEFAULT. Unspecified ranges will be automatically determined using the function GetDualRangeN.

`m` An (optional) number of points to use per list of points per dimension. Over-ridden if `rxi` is an array or list.

Output: The conjugate, as an NPWF object.

A.2.2.7 CCAT[ConvexN]

Description: Tests a given NPWF object for convexity. Tests to a slightly lower numerical precision than that used in the calculations of the object. Thus, for small non-convexities (of a magnitude on the same order as the precision in use), this routine may return incorrect results.

Usage:

```
ConvexN( npwf )
```

Input:

`npwf`	An `NPWF` object.

Output: A boolean value.

Bibliography

[1] L. Ahlfors. *Complex Analysis.* McGraw-Hill, New York, 1966.

[2] H.H. Bauschke and M. v. Mohrenschildt. Fenchel conjugates and subdifferentiation in Maple. Technical Report CORR 97-23, Department of Combinatorics and Optimization, University of Waterloo, 1997.

[3] H.H. Bauschke and M. v. Mohrenschildt. Symbolic computation of Fenchel conjugates. To appear in ACM SIGSAM bulletin, 2005.

[4] J.M. Borwein and A.S. Lewis. *Convex Analysis and Nonlinear Optimization.* Springer-Verlag, New York, 2000.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, UK, 2004.

[6] Y. Brenier. Un algorithme rapide pour le calcul de transformée de Legendre-Fenchel discrètes. *C. R. Acad. Sci. Paris Sér. I Math*, 308:587–589, 1989.

[7] L. Corrias. Fast Legendre-Fenchel transform and applications to Hamilton-Jacobi equations and conservation laws. *SIAM Journal of Numerical Analysis*, 33:1534–1558, 1996.

[8] A.R. De Pierro and A.N. Iusem. A relaxed version of Bregman's method for convex programming. *Journal of Optimization Theory and Applications*, 51(3):421–440, 1986.

[9] Hiriart-Urruty J.B. Lipschitz r-continuity of the approximate subdifferential of a convex function. *Math. Scand.*, 47:123–134, 1980.

[10] D.E. Knuth. *The Art of Computer Programming, Vol. 3: Sorting and Searching.* Addison-Wesley, Reading, MA, 1973.

[11] Y. Lucet. A fast computational algorithm for the Legendre-Fenchel transform. *Computational Optimization and Applications*, 6(1):27–57, 1996.

[12] Y. Lucet. Faster than the fast Legendre transform, the linear-time Legendre transform. *Numerical Algorithms*, 16:171–185, 1997.

[13] D. Luenberger. *Optimization by Vector Space Methods.* Wiley, New York, 1969.

[14] A. Noullez and M. Vergassola. A fast Legendre transform algorithm and applications to the adhesion model. *Journal of Scientific Computing*, 9:259–281, 1994.

[15] F.P. Preparata and M.I. Shamos. *Computational Geometry.* Springer, Berlin, 1990.

[16] R.T. Rockafeller. *Convex Analysis.* Princeton University Press, Princeton, NJ, 1970.

[17] W.L. Winston. *Introduction to Mathematical Programming.* Duxbury Press, Belmont, CA, 1995.