Symbolic computation of Fenchel conjugates

Heinz H. Bauschke*
Department of Mathematics,
Irving K. Barber School
UBC Okanagan,
Canada
heinz.bauschke@ubc.ca
and
Martin v. Mohrenschildt
Computing and Software
McMaster University
1280 Main Street West,
Hamilton, Ontario L8S 4L7
Canada
mohrens@mcmaster.com

Abstract

Convex optimization deals with certain classes of mathematical optimization problems including least-squares and linear programming problems. This area has recently been the focus of considerable study and interest due to the facts that convex optimization problems can be solved efficiently by interior-point methods and that convex optimization problems are actually much more prevalent in practice that previously thought.

Key notions in convex optimization are the Fenchel conjugate and the subdifferential of a convex function. In this paper, we build a new bridge between convex optimization and symbolic mathematics by describing the Maple package `fenchel`, which allows for the symbolic computation of these objects for numerous convex functions defined on the real line. We are able to symbolically reproduce computations for finding Fenchel conjugates and subdifferentials for numerous nontrivial examples found in the literature.

1 Motivation

1.1 Definitions and Basic Results

1.1.1 Convex function

Suppose f is a function defined on $\mathbb{R}^n$ which takes on values in $]-\infty,+\infty] = \mathbb{R}\cup\{+\infty\}$. Recall that f is *convex* if

$$f(\lambda_1 x_1 + \lambda_2 x_2) \leq \lambda_1 f(x_1) + \lambda_2 f(x_2),$$

for every $x_1, x_2 \in \mathbb{R}^n$, and all $\lambda_1, \lambda_2 > 0$ with $\lambda_1 + \lambda_2 = 1$. The *(effective) domain of f*, written $\operatorname{dom} f$, is the set of all points where f is not $+\infty$. Convex functions lie at the heart of convex, functional and real analysis as well as convex optimization. Several excellent monographs on the subject are available: Rockafellar's classical [17], Roberts and Varberg's gentler [16], and the more recent works by Hiriart-Urruty and Lemaréchal [9, 10], by Borwein and Lewis [5], and by Boyd and Vandenberghe [6]. We point the reader to these references for further information.

*H. H. Bauschke's work was partially supported by the Natural Sciences and Engineering Research Council of Canada.

1.1.2 Subdifferential and subgradient

In calculus we learn that a minimizer of f, say $\bar{x}$, is necessarily a critical point: $\nabla f(\bar{x}) = 0$. Since several interesting convex functions are *not* everywhere differentiable, this technique is not immediately available. Instead, one defines the so-called *subdifferential of f at x* by

$$\partial f(x) := \{y \in \mathbb{R}^n : \langle y, x' - x\rangle \le f(x') - f(x),\ \forall x' \in \mathbb{R}^n\}.$$

Here and elsewhere in the paper, $\langle \cdot,\cdot\rangle$ denotes the standard inner product on $\mathbb{R}^n$. Also, the symbol $\partial f(x)$ is standard in convex analysis — it should not be confused with partial derivatives. Members of the subdifferential are called *subgradients*. This idea generalizes differentiability: indeed, f is differentiable at x if and only if $\partial f(x) = \{\nabla f(x)\}$. The importance of subgradients in optimization stems from the fact that

$$\bar{x} \text{ is a (global) minimizer of } f \Leftrightarrow 0 \in \partial f(\bar{x}).$$

If f is convex and defined on $\mathbb{R}$, then the left (f'_-) and right (f'_+) derivatives exist at every point in $\operatorname{dom} f$; moreover, the subdifferential is a closed interval completely described by these directional derivatives:

$$\partial f(x) = [f'_-(x), f'_+(x)], \quad \forall x \in \operatorname{dom} f.$$

1.1.3 Fenchel conjugate

The *Fenchel conjugate* (also known as the Fenchel-Legendre transform) of f, denoted f^*, is defined by

$$f^*(y) = \sup_{x\in\mathbb{R}^n} \big[\langle y,x\rangle - f(x)\big], \quad \forall y \in \mathbb{R}^n;$$

it is always a convex and lower semi-continuous function on $\mathbb{R}^n$. (Recall that a function g is called lower semi-continuous at y, if $\underline{\lim}_{y'\to y}\, g(y') \ge g(y)$.) The role of the Fenchel conjugate in convex analysis is important; it can be compared to the role of the Fourier transform in harmonic analysis. Assuming lower-semicontinuity, the operation of performing Fenchel conjugation twice recovers the original function. In fact, more is true:

$$f \text{ is convex and lower semi-continuous} \Leftrightarrow f = f^{**}.$$

An immediate consequence of the definition of the Fenchel conjugate is the famous *Fenchel-Young inequality:*

$$\text{(FYI)} \qquad f(x) + f^*(y) \ge \langle x,y\rangle, \quad \forall x,y \in \mathbb{R}^n.$$

The *Fenchel-Young equality* is $f(x)+f^*(y) = \langle x,y\rangle$. Sufficient and necessary conditions for this equality to hold are formulated in terms of subgradients: for $x,y \in \mathbb{R}^n$, we have

$$\text{(FYE)} \qquad f(x)+f^*(y) = \langle x,y\rangle \Leftrightarrow y \in \partial f(x) \Leftrightarrow x \in \partial f^*(y).$$

1.2 Convex Optimization Duality

Convex optimization deals with optimization problems of the following type:

$$\text{(P)} \qquad p = \inf_{x\in\mathbb{R}^n} \big[f(x) + g(Ax)\big],$$

where $A \in \mathbb{R}^{m\times n}$ and f (resp. g) is convex and lower semi-continuous on $\mathbb{R}^n$ (resp. $\mathbb{R}^m$). The program (P) is called the *primal problem*. The problem *dual* to (P) is defined by

$$\text{(D)} \qquad d = -\inf_{z\in\mathbb{R}^m} \big[f^*(-A^*z) + g^*(z)\big],$$

where A^* is the transpose of A.

We now turn to the fundamental results on this primal-dual pair of optimization problems.

Fact (Fenchel's Duality Theorem) Suppose $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^m$. Then:
Weak duality: $p \ge d$.
Karush-Kuhn-Tucker conditions: x solves (P), z solves (D), and $p = d$ if and only if

$$Ax \in \partial g^*(z) \text{ and } -A^*z \in \partial f(x).$$

Strong duality: If $A(\operatorname{dom} f) \cap \operatorname{int}(\operatorname{dom} g) \neq \emptyset$ or a similar constraint qualification is satisfied[1], then $p = d$ and the infimum defining d is attained.
Primal solutions: If z is a solution of (D), then the solutions of (P) are equal to the (possibly empty) set

$$A^{-1}\partial g^*(z) \cap \partial f^*(-A^*z).$$

Fenchel's Duality Theorem clearly shows the importance of subdifferentials and Fenchel conjugates.

As an aside, we point out that it is possible to recover the well-known *Linear Programming Duality* from (a variant of) Fenchel's Duality Theorem: consider, for instance, the following primal linear program:

$$\text{(primal LP)} \qquad \min\big\{\langle c,x\rangle : x \in \mathbb{R}^n, x \geq 0, Ax = b\big\},$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m\times n}$. To put this into the framework of the Fenchel's Duality Theorem, we first set $f(x) = \langle c,x\rangle$, if $x \geq 0$; $f(x) = +\infty$, otherwise. This f is convex and lower semi-continuous on $\mathbb{R}^n$. Secondly, we define $g(y) = 0$, if $y = b$; $g(y) = +\infty$, otherwise. Here g is convex and lower semi-continuous on $\mathbb{R}^m$. With these definitions, (primal LP) corresponds precisely to (P). Next, one easily computes that $f^*(x) = 0$ if $x \leq c$; $f^*(x) = +\infty$ otherwise, and that $g^*(y) = \langle y,b\rangle$. Then it is not hard to see that the Fenchel dual of (primal LP) is

$$\text{(dual LP)} \qquad \max\big\{\langle y,b\rangle : y \in \mathbb{R}^m, A^*y \leq c\big\},$$

which is *precisely* the dual program in the sense of Linear Programming.

1.3 Applications

A convex function f defined on $\mathbb{R}^n$ is called *separable*, if it can be written as

$$f(x) = \sum_{j=1}^{n} f_j(x_j), \quad \forall x \in \mathbb{R}^n,$$

where each f_j is a convex function on $\mathbb{R}$. For such a function f, we have

$$\partial f(x) = \partial f_1(x_1) \times \cdots \times \partial f_n(x_n), \quad \forall x \in \mathbb{R}^n$$

and

$$f^*(y) = \sum_{j=1}^{n} f_j^*(y_j), \quad \forall y \in \mathbb{R}^n.$$

Consequently, computing subdifferentials and conjugates of separable convex functions is (essentially) no harder than dealing with one-dimensional convex functions. We state two areas of applications.

1.3.1 Separable convex optimization

Separable convex programs are optimization problems with separable objective functions. This includes *least squares* and *linear programming* problems as well as many problems arising in applications (for a problem arising in *network optimization* see [4, Section 5.4.2]).

The study of *penalty* and *proximal point-like methods* for solving convex programs relies implicitly on Fenchel conjugates. Separable penalty functions are often used in practice; see for instance [4, Sections 5.4.5 and 5.4.6] and [15].

1.3.2 Inequalities

An important source for *inequalities in classical analysis* is the Fenchel-Young inequality and its characterization of equality; see [14, Sections XIV.9 and XXVI.4].

1.4 Aim of this paper

The preceding sections show that it is quite worthwhile and interesting to be able to compute Fenchel conjugates and subdifferentials, even in the one-dimensional case.

In this paper, we aim to build a new bridge between convex optimization and symbolic computation. We present the Maple package `fenchel`*, which allows the symbolic computation of Fenchel conjugates and subdifferentials for many convex functions defined on the real line.*

[1] We write $\operatorname{int}(\operatorname{dom} g)$ for the *interior* of $\operatorname{dom} g$.

1.5 The Maple package **fenchel**

It is tempting to try to compute Fenchel conjugates in Maple by using `piecewise` or `maximize`. Unfortunately, neither is well suited for this task: On the one hand, the normal form of a `piecewise`-function taking the value $+\infty$ may contain terms of the form $\infty - \infty$, which results in `undefined`. (See [18] and [19] for more on `piecewise` and its underlying theory.) On the other hand, the command `maximize` is not able to handle many of the examples listed below. We thus were led to create the package `fenchel` to conjugate and subdifferentiate convex functions. It is the first package that is able to symbolically manipulate convex functions in this generality; the code (for Maple 9) is available from the authors upon request. This paper is an extension of the technical report [3]. (We refer the reader to [13] for the complementary aspect of computing Fenchel conjugates *numerically*.)

1.6 A good class of functions

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

(i) f is a function from $\mathbb{R}$ to $]-\infty, +\infty]$;
(ii) f is convex;
(iii) f is continuous on its effective domain;
(iv) and there are finitely many points
$$x_0 = -\infty < x_1 < \cdots < x_{m-1} < x_m = +\infty$$
such that f restricted to each open interval $]x_{i-1}, x_i[$ is one of the following: identically equal to $+\infty$, or affine, or differentiable with strictly increasing derivative.

Since we deal with functions on the real line, condition (iii) is actually equivalent to the lower semi-continuity of f. If f_1, f_2 belong to $\mathcal{F}$ and $\alpha_1, \alpha_2 \geq 0$, then $\alpha_1 f_1 + \alpha_2 f_2$ and f_1^* are in $\mathcal{F}$ as well. The class $\mathcal{F}$ is thus very well-suited for our purpose.

1.7 Functionality of the package **fenchel**

New commands: `cf` and `sd` (entering convex functions and subdifferentials); `cfplot` and `sdplot` (plotting); `cfeval` and `sdeval` (evaluating at a point); `subdiff` and `invert` (computing and inverting the subdifferential of a convex function); `conj` (computing the Fenchel conjugate); `convpw` (converts from `cf` to `piecewise`).
Supported Maple commands The following Maple commands work for `cf`-functions: `convert`, `evalf`, `expand`, `latex`, `normal`, `print`, `simplify`, `type`. Most useful to us were `simplify` and `convert` (from `piecewise` to `cf`).

1.8 Algorithmic approach

A convex lower semi-continuous function on the real line is very well-behaved; for instance, it is subdifferentiable on the interior of its domain. This allows the computation of the Fenchel conjugate at such an interior point y in two steps: firstly, solve $y \in \partial f(x)$ for x — this is the key step — and let $\bar{x}$ be such a solution. Secondly, use the Fenchel-Young equality (see equation (FYE) in Section 1.1.3) to obtain $f^*(y) = \langle \bar{x}, y \rangle - f(\bar{x})$. Continuity then determines the value of the Fenchel conjugate at boundary points.

2 A dozen examples

Let us now illustrate the functionality of `fenchel` on a representative set of interesting convex functions.

2.1 The notorious absolute value

Perhaps the most simple nondifferentiable convex function is the absolute value function: $f = |\cdot|$. Its derivative at 0 fails to exist, since $f'_-(0) = -1 < 1 = f'_+(0)$. Accordingly, the subdifferential of f at 0 is $\partial f(0) = [-1, 1]$. The conjugate of f, f^*, is the *indicator* function of the interval $[-1, 1]$: $f^*(y) = 0$, if $y \in [-1, 1]$; $f^*(y) = \infty$, otherwise. After generating the `cf` representation of the absolute value function via `f1 := convert(abs(x),cf)`,[2] we compute `subdiff(f1)` and `g1 := conj(f1,y)`, which results in the following output:

$$\begin{cases} \{-1\} & (-\infty < x) \textbf{ and } (x < 0) \\ [-1,1] & x = 0 \\ \{1\} & (0 < x) \textbf{ and } (x < \infty) \end{cases}$$

[2] Here and elsewhere in the paper, we occasionally suppress colons and semicolons in Maple commands for better readability.

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

2.2 The negative entropy

The exponential function together with the negative (or Boltzmann-Shannon) entropy is one of the most famous pairs of Fenchel conjugates. We notice that the Fenchel conjugate can take the value $+\infty$, even if the original function is very well-behaved:

$$f(x) = \exp(x) \text{ and } f^*(y) = \begin{cases} +\infty, & \text{if } y < 0; \\ 0, & \text{if } y = 0; \\ y\ln(y) - y, & \text{otherwise.} \end{cases}$$

In `fenchel`, it suffices to enter `f2 := convert(exp(x),cf)` followed by `g2 := conj(f2,y)` to achieve:

$$f2 := \begin{cases} e^x & (-\infty < x) \textbf{ and } (x < \infty) \end{cases}$$

$$g2 := \begin{cases} \infty & (-\infty < y) \textbf{ and } (y < 0) \\ 0 & y = 0 \\ y(\ln(y) - 1) & (0 < y) \textbf{ and } (y < \infty) \end{cases}$$

2.3 De Pierro and Iusem's example

This pair was suggested by De Pierro and Iusem on page 438 of [8]; it was also utilized in [2]. We enter the function directly via `cf`:

```
f3 := cf([[-infinity,1/2*(x^2-4*x+3),1],
          [1,0,1],[1,-ln(x),infinity]],x);
```

$$f3 := \begin{cases} 1/2x^2 - 2x + 3/2 & (-\infty < x) \textbf{ and } (x < 1) \\ 0 & x = 1 \\ -\ln(x) & (1 < x) \textbf{ and } (x < \infty) \end{cases}$$

The conjugate of $f3$ is now found via `g3 := conj(f3,y)`:

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

2.4 Affine and quadratic functions

An *affine* function is a function of the form $x \mapsto bx + c$, where b and c are real constants. Conjugates of affine functions are finite only at one point: `conj(convert(b*x+c,cf,x),y)` results in

$$\begin{cases} \infty & (-\infty < y) \textbf{ and } (y < b) \\ -c & y = b \\ \infty & (b < y) \textbf{ and } (y < \infty) \end{cases}$$

On the other hand, the class of quadratic convex functions is closed under Fenchel conjugation: the commands `assume(a>0)` and `conj(convert(a*x^2+b*x+c,cf,x),y)` lead to [3]

$$\begin{cases} \dfrac{y^2 - 2yb + b^2 - 4ca}{4a} & (-\infty < y) \textbf{ and } (y < \infty) \end{cases}$$

[3] We assume here `interface(showassumed=0)`.

The function $\frac{1}{2}x^2$ is special since it is the only quadratic function that coincides with its Fenchel conjugate. A more general result states that the only self-conjugate convex lower semi-continuous function on $\mathbb{R}^n$ is $x \mapsto \frac{1}{2}\|x\|^2$.

2.5 Lagarias and Weiss's function

One of the most famous open problems in mathematics is the so-called $3x+1$ *problem*:

> Define a self map T of the positive integers by
>
> $$T(n) := \begin{cases} n/2, & \text{if } n \text{ is even;} \\ 3n+1, & \text{if } n \text{ is odd.} \end{cases}$$
>
> Is it true repeated application of T leads eventually to 1, for every positive integer n?

See [11] for an excellent survey. In [12], Lagarias and Weiss study a *random walk model* for the $3x+1$ problem. They are led to consider the function $f(x) = \frac{1}{2}(2^x + (\frac{2}{3})^x)$ and its conjugate $g = f^*$. For instance, they prove that the open interval $]\ln(2/3), \ln(2)[$ is contained in $\operatorname{dom} g$, that $g(\frac{1}{2}\ln(\frac{4}{3})) = 0$, and that $g(y) = y$ has a unique solution $\bar{y} \approx 0.02399$. (The quantity $\bar{y}$ is important in their analysis.) We can at least illustrate their findings. In fact, `conj` returns a closed form for g, which yields $\operatorname{dom} g = [\ln(2/3), \ln(2)]$ and g takes the value $\ln(2)$ at both endpoints of the interval. Because of its length, we do not list the closed form of g. We first show that $g(\frac{1}{2}\ln(\frac{4}{3})) = 0$ by using `fenchel`'s evaluation command `cfeval`. We then confirm Lagarias and Weiss's numerical approximation of $\bar{y}$ using `op` and `fsolve`. More concretely, the cascade of commands

```
f5 := convert(ln((1/2)*(2^x+(2/3)^x)),cf):
g5 := conj(f5,y):cfeval(g5,y=1/2*ln(4/3));
temp := op([1,3,2],g5):ybar := fsolve(temp=y,y);
```

yields

$$0$$

$$ybar := .02399367662$$

2.6 A barrier function

In [15], Polyak and Teboulle study convex optimization algorithms and they use (implicitly) convex functions and their Fenchel conjugates. (More precisely, the authors consider *concave* functions and correspondingly *concave conjugates*, which is mathematically equivalent to the present convex setting.) Let us recover the last example in their paper. We also utilize `factor` to simplify the Fenchel conjugate. (The commands `expand` and `normal` can be used similarly.)

```
f6 := cf([[-infinity,exp(-4*x-2),-1/2],
          [-1/2,1,-1/2],
          [-1/2,-x/(1+x),infinity]],x);
```

$$f6 := \begin{cases} e^{-4x-2} & (-\infty < x) \textbf{ and } (x < -1/2) \\ 1 & x = -1/2 \\ -\frac{x}{1+x} & (-1/2 < x) \textbf{ and } (x < \infty) \end{cases}$$

```
factor(conj(f6,y));
```

$$\begin{cases} 1/4\,y\,(-1+2\ln(2)-\ln(-y)) & (-\infty < y) \textbf{ and } (y < -4) \\ 1 & y = -4 \\ -\frac{(\sqrt{-y}-1)(y+\sqrt{-y})}{\sqrt{-y}} & (-4 < y) \textbf{ and } (y < 0) \\ 1 & y = 0 \\ \infty & (0 < y) \textbf{ and } (y < \infty) \end{cases}$$

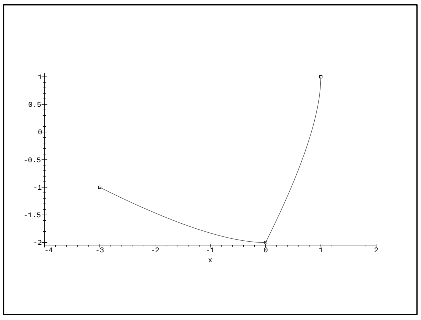

Figure 1: The graph of $f8$

2.7 An example from Bertsekas's book

The package `fenchel` is powerful enough to recover all conjugate pairs in Bertsekas's book [4]; for instance, let us consider the following paring from page 468:

```
f7 := convert(piecewise(x>=0, a*x+x^2,0),cf,x);
g7 := factor(conj(f7,y));
```

$$f7 := \begin{cases} 0 & (-\infty < x) \textbf{ and } (x < 0) \\ 0 & x = 0 \\ a\,x + x^2 & (0 < x) \textbf{ and } (x < \infty) \end{cases}$$

$$g7 := \begin{cases} \infty & (-\infty < y) \textbf{ and } (y < 0) \\ 0 & y = 0 \\ 0 & (0 < y) \textbf{ and } (y < a) \\ 0 & y = a \\ 1/4\,(y-a)^2 & (a < y) \textbf{ and } (y < \infty) \end{cases}$$

2.8 An example from Rockafellar's text

The next function can be found on page 229 in Rockafellar's seminal [17]. We must enter it via `cf`, since `piecewise` cannot properly handle two occurrences of ∞. (See our comments in Section 1.5.)

```
f8 := convert(piecewise(-3<=x and x<=1,
       abs(x)-2*sqrt(1-x), infinity),cf);
```

$$f8 := \begin{cases} \infty & (-\infty < x) \textbf{ and } (x < -3) \\ -2\sqrt{4}+3 & x = -3 \\ -2\sqrt{1-x} - x & (-3 < x) \textbf{ and } (x < 0) \\ -2 & x = 0 \\ -2\sqrt{1-x} + x & (0 < x) \textbf{ and } (x < 1) \\ 1 & x = 1 \\ \infty & (1 < x) \textbf{ and } (x < \infty) \end{cases}$$

We now use

```
cfplot(f8,x=-4..2,scaling=constrained,
       axes=framed)
```

to plot the function (see Figure 1).

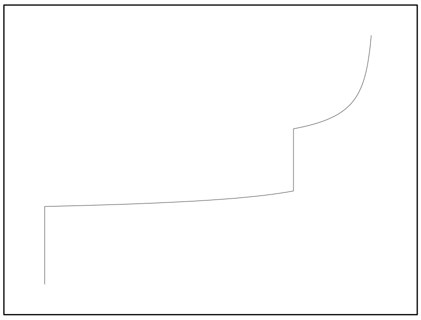

Figure 2: The subdifferential of $f8$

Next, to find and plot its subdifferential, we use `sdf8 := subdiff(f8)` followed by `sdplot(sdf8,x=-3..1,view=[-3..1,-3..5]` which yields

$$sdf8 := \begin{cases} \{\} & (-\infty < x) \textbf{ and } (x < -3) \\ [-\infty, -1/2] & x = -3 \\ \{-\frac{-1+\sqrt{1-x}}{\sqrt{1-x}}\} & (-3 < x) \textbf{ and } (x < 0) \\ [0,2] & x = 0 \\ \{\frac{1+\sqrt{1-x}}{\sqrt{1-x}}\} & (0 < x) \textbf{ and } (x < 1) \\ \{\} & x = 1 \\ \{\} & (1 < x) \textbf{ and } (x < \infty) \end{cases}$$

and the plot in Figure 2. Finally, we find the conjugate of $f8$, convert it to `piecewise` for further simplification: `g8 := conj(f8,y); simplify(convpw(g8));`

$$\begin{cases} -3y+1 & (-\infty < y) \textbf{ and } (y < -1/2) \\ 5/2 & y = -1/2 \\ \frac{y^2+2y+2\,\mathrm{signum}(y+1)}{y+1} & (-1/2 < y) \textbf{ and } (y < 0) \\ 2 & y = 0 \\ 2 & (0 < y) \textbf{ and } (y < 2) \\ 2 & y = 2 \\ \frac{y^2-2y+2\,\mathrm{signum}(y-1)}{y-1} & (2 < y) \textbf{ and } (y < \infty) \end{cases}$$

$$\begin{cases} -3y+1 & y \le -1/2 \\ \frac{y^2+2y+2}{y+1} & y \le 0 \\ 2 & y \le 2 \\ \frac{y^2-2y+2}{y-1} & 2 < y \end{cases}$$

2.9 An infimal convolution

Given two convex lower semi-continuous functions f and h, the function $(f^* + h^*)^*$ is called (the closure of) the *infimal convolution* of f and h. If one of the functions is differentiable, then so is their infimal convolution; hence this operation is a *regularization*. In the next example, we regularize the (nondifferentiable) absolute value function $|x|$ with $\frac{1}{2}x^2$: `f9 :=  convert(1/2*x^2,cf):`

```
conj(simplify(conj(f1,y) + conj(f9,y)),x);
```

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

2.10 An occurrence of the Lambert W function

The following pair is from [5, Exercise 3.3.12.(e)]. The function $x \mapsto \frac{1}{2}|x|^2 + x\ln(x) - x$ underlies the log-quad methods introduced by Auslender *et al.* [1]. We compute its conjugate as follows.

```
f10 := simplify(f9 + subs(y=x,g2)):
g10 := convpw(conj(f10,y));
```

$$g10 := \tfrac{1}{2}\mathit{LambertW}\left(e^y\right)\left(\mathit{LambertW}\left(e^y\right)+2\right)$$

Here, *LambertW* stands for Lambert's W-function; see [7] for a nice survey.

2.11 The negative logarithm

The function $x \mapsto -\ln(x)$ lies at the heart of modern *interior point methods* in Linear Programming. To find its conjugate, we enter

```
f11 := cf([[-infinity,infinity,0],[0,infinity,0],
          [0,-ln(x),infinity]],x)
```

$$f11 := \begin{cases} \infty & (-\infty < x) \textbf{ and } (x < 0) \\ \infty & x = 0 \\ -\ln(x) & (0 < x) \textbf{ and } (x < \infty) \end{cases}$$

and `g11 := conj(f11,y)`

$$g11 := \begin{cases} -1-\ln(-y) & (-\infty < y) \textbf{ and } (y < 0) \\ \infty & y = 0 \\ \infty & (0 < y) \textbf{ and } (y < \infty) \end{cases}$$

Note that $g11(y) = -1 + f11(-y)$; in this sense, $f11$ is self-conjugate up to reflection and an additive constant.

2.12 A case of Young's inequality

Suppose $1 < p < +\infty$ and let q be given by $\frac{1}{p} + \frac{1}{q} = 1$. The inequality

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

is often referred to as *Young's inequality*. It can be used to give an easy proof of *Hölder's inequality*. In fact, since

$$\left(\tfrac{1}{p}|\cdot|^p\right)^* = \tfrac{1}{q}|\cdot|^q,$$

Young's is a special case of the Fenchel-Young inequality (FYI) from Section 1.1.3. In our last example, we derive the conjugate pair for $p = 3$:

```
f12 := convert(1/3*abs(x)^3,cf);
```

$$f12 := \begin{cases} -1/3x^3 & (-\infty < x) \textbf{ and } (x < 0) \\ 0 & x = 0 \\ 1/3x^3 & (0 < x) \textbf{ and } (x < \infty) \end{cases}$$

```
g12 := conj(f12,y);
```

$$g12 := \begin{cases} 2/3\,(-y)^{3/2} & (-\infty < y) \textbf{ and } (y < 0) \\ 0 & y = 0 \\ 2/3y^{3/2} & (0 < y) \textbf{ and } (y < \infty) \end{cases}$$

3 Limitations

The biggest challenge for symbolically finding Fenchel conjugates is to invert the Fenchel-Young equality (see equation (FYE) in Section 1.1.3). We rely on the Maple function `solve`, which by its nature has to deal with branch cuts and hence does not always allow a closed form symbolic inverse. A typical example is $f(x) = \frac{1}{4}x^4$. Maple correctly finds three solutions of $\nabla f(x) = y$, *i.e.*, three cubic roots of y. But none of these expression is the real root for *all* values of y. In essence, this is why we cannot recover the p-powers discussed in Section 2.12.

4 Concluding remarks

In `fenchel`, we have implemented Fenchel conjugation and subdifferentials for convex functions on the real line. We have presented well-working examples and commented on limitations of the code. We hope that

- `fenchel` will be useful to both researchers and instructors in convex optimization and analysis; and that
- `fenchel` will spark further activities in this area resulting perhaps in code that tackles nonseparable multi-dimensional convex functions.

References

[1] A. Auslender, M. Teboulle, and S. Ben-Tiba. A logarithmic-quadratic proximal method for variational inequalities. *Computational Optimization and Applications*, 12:31–40, 1999.

[2] H.H. Bauschke and J.M. Borwein. Legendre functions and the method of random Bregman projections. *Journal of Convex Analysis*, 4(1):27–67, 1997.

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

[4] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

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

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

[7] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, and D.E. Knuth. On the Lambert W Function. *Advances in Computational Mathematics*, 5:329–359, 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, December 1986.

[9] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*, volume 305 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1993.

[10] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*, volume 306 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1993.

[11] J.C. Lagarias. The $3x+1$ problem and its generalizations. *American Mathematical Monthly*, 92:3–23, 1985.

[12] J.C. Lagarias and A. Weiss. The $3x+1$ problem: two stochastic models. *The Annals of Applied Probability*, 2(1):229–261, 1992.

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

[14] D.S. Mitrinović, J.E. Pečarić, and A.M. Fink. *Classical and New Inequalities in Analysis*. Kluwer, 1993.

[15] R. Polyak and M. Teboulle. Nonlinear rescaling and proximal-like methods in convex optimization. *Mathematical Programming*, 76:265–284, 1997.

[16] A.W. Roberts and D.E. Varberg. *Convex Functions*. Academic Press, New York, 1973.

[17] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

[18] M. v. Mohrenschildt. Using "piecewise" to solve classes of control theory problems. *MapleTech*, 4(3):33–37, 1997.

[19] M. v. Mohrenschildt. A normal form for function rings of piecewise functions. *Journal of Symbolic Computation*, 26:607–619, 1998.