Fenchel conjugates and subdifferentials in Maple

***Heinz H. Bauschke**[*]**and Martin von Mohrenschildt**[†]*

Abstract: The notions of the Fenchel conjugate and the subdifferential of a convex function are fundamental in optimization. The package `fenchel` allows the symbolic computation of these objects for numerous convex functions defined on the real line.

Keywords: Convex function, Fenchel conjugate, Legendre-Fenchel transform, Legendre transform, subdifferential, subgradient.

Motivation

DEFINITIONS

Convex function

Suppose f is a function defined on $\mathbb{R}^n$ with values in $(-\infty, +\infty]$. 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. Several excellent monographs on the subject are available: Rocakfellar's classical [13], Roberts and Varberg's gentler [12], and Hiriart-Urruty and Lemaréchal's recent [5, 6]. In these references, the reader may find further information and proofs of the results stated below.

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 various of the most interesting convex functions are *not* everywhere differentiable, this technique is not immediately available. Instead, one defines the so-called *subdifferential of* f *at* x, denoted $\partial f(x)$, by

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

(Here and elsewhere in the paper, $\langle\cdot,\cdot\rangle$ denotes the standard inner product on $\mathbb{R}^n$.) 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 derviatives:

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

Fenchel conjugate

The *Fenchel conjugate* 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 $\underline{\lim}_{y'\to y} g(y') \geq g(y)$.) Assuming lower-semicontinuity, the operation of performing Fenchel conjugation twice recovers the original function. In fact, more is true:

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

Immediate from the definition of the Fenchel conjugate is the famous *Fenchel/Young inequality:*

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

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

$$f(x) + f^*(y) = \langle x, y\rangle \Leftrightarrow y \in \partial f(x) \Leftrightarrow x \in \partial f^*(y).$$

CONVEX OPTIMIZATION DUALITY

Convex optimization studies convex programs 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 semicontinuous on $\mathbb{R}^n$ (resp. $\mathbb{R}^m$). The program (P) is called the *primal problem*. The problem *dual* to (P) is defined by

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

[*]Department of Mathematics and Statistics, Okanagan University College, Kelowna, British Columbia V1V 1V7, Canada. `bauschke@cecm.sfu.ca`. This research was supported by an NSERC Postdoctoral Fellowship and by the Department of Combinatorics and Optimization at the University of Waterloo.

[†]Department of Computing And Software, Faculty of Engineering, McMaster University, Hamilton, Ontario L8S 4L8, Canada. `mohrens@ece.eng.mcmaster.ca`.

where A^* is the transpose of A.

We collect the fundamental results on this primal-dual pair of optimization problems in the following.

Fact (Fenchel's Duality Theorem) Suppose $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^m$. Then:

Weak duality: $p \geq d$.

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 $A(\operatorname{dom} f) \cap \operatorname{int}(\operatorname{dom} g) \neq \emptyset$, 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 Fenchel's Duality Theorem, we first set $f(x) = \langle c, x\rangle$, if $x \geq 0$; $f(x) = +\infty$, otherwise. Then f is convex and lower semi-continuous on $\mathbb{R}^n$. Secondly, we let $g(y) = 0$, 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!

One-dimensional 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.

Separable convex optimizaton

Separable convex programs are optimization problems with separable objective functions. This includes the problem of *network optimzation*; see [3, 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 [3, Sections 5.4.5 and 5.4.6]) and [11].

Inequalities

An important source for *inequalities in classical analysis* is the Fenchel/Young inequality and its characterization of equality; see [10, Sections XIV.9 and XXVI.4].

Aim of this paper

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

Our objective in this paper is the presentation of the package `fenchel`, *which allows the symbolic computation of Fenchel conjugates and subdifferentials for many convex functions defined on the real line.*

Why `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 an error. (See [14] and [15] 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 `fenchel` to conjugate and subdifferentiate convex functions. As far as we know, it is the first package able to manipulate convex functions in this generality.

Good algorithms for computing Fenchel conjugates *numerically* are available, see Lucet's [9] for further information.

Obtaining `fenchel`

The Centre for Experimental and Constructive Mathematics (CECM) maintains a *Computational Convex Analysis* page at `cecm.sfu.ca/projects/CCA`; the reader can find a link to `fenchel` from there.

A GOOD CLASS OF FUNCTIONS

Let $\mathcal{F}$ be the class of all functions f satisfying the conditions listed next:
(i) f is a function from $\mathbb{R}$ to $(-\infty, +\infty]$.
(ii) f is convex.
(iii) f is continuous on its effective domain.
(iv) 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.

Note that — because we deal with functions on the real line — condition (iii) is 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. (It is possible that $\max\{f_1, f_2\}$ lies outside $\mathcal{F}$, but we do not discuss the maximum operation here.)

The class $\mathcal{F}$ is thus very well-suited for our purpose.

WHAT CAN `FENCHEL` DO?

We classify `fenchel`'s procedures as follows.

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`, `normal`, `print`, `simplify`, `type`. Most useful to us were `simplify` and `convert` (from `piecewise` to `cf`).

Rather than formally discussing `fenchel`'s commands, we prefer to demonstrate them by examples.

HOW DOES `FENCHEL` WORK?

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 to obtain $f^*(y) = \langle \bar{x}, y\rangle - f(\bar{x})$. Continuity then determines the value of the Fenchel conjugate at boundary points.

Ten examples

1. THE NOTORIOUS ABSOLUTE VALUE

Perhaps the simplest 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. We use the commands `subdiff` and `conj` to rederive this. The `convert` command makes entering the absolute value very easy.

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

$$f1 := \begin{cases} -x & (-\infty < x) \textbf{ and } (x < 0) \\ 0 & x = 0 \\ x & (0 < x) \textbf{ and } (x < \infty) \end{cases}$$

```
> subdiff(f1);
```

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

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

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

2. THE BOLTZMANN/SHANNON ENTROPY

The exponential function together with the Boltzmann/Shannon entropy is one of the most famous pairs of Fenchel conjugates. It shows that the Fenchel conjugate can take the value ∞, 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`, we recapture this computation as follows:

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

$$f2 := \{ e^x \quad (-\infty < x) \textbf{ and } (x < \infty)$$

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

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

```
> conj(g2,x);
```

$$\{ e^x \quad (-\infty < x) \textbf{ and } (x < \infty)$$

3. De Pierro and Iusem's example

This pair was suggested by De Pierro and Iusem on page 438 of [4]; it was quite useful in [1] and [2]. This time, we enter the function directly via `cf` (`convert` would have worked, too).

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

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

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

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

4. Affine and quadratic functions

An *affine* function is of the form $x \mapsto bx + c$, where b, c are real constants. Conjugates of affine functions are everywhere ∞, except at one point.

```
> convert(b*x+c,cf,x);
```

$$\{\, b\,x + c \quad (-\infty < x) \textbf{ and } (x < \infty)$$

```
> conj(%,y);
```

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

On the other hand, the class of quadratic convex functions is closed under Fenchel conjugation:

```
> assume(a>0);
> convert(a*x^2+b*x+c,cf,x);
```

$$\{\, a\tilde{}\, x^2 + b\,x + c \quad (-\infty < x) \textbf{ and } (x < \infty)$$

```
> conj(%,y);
```

$$\left\{ \frac{1}{4}\frac{y^2 - 2\,y\,b + b^2 - 4\,c\,a\tilde{}}{a\tilde{}} \quad (-\infty < y) \textbf{ and } (y < \infty) \right.$$

The function $\frac{1}{2}x^2$ is special since it is the only quadratic that coincides with its Fenchel conjugate. In fact, much more is true: the only self-conjugate convex lower semi-continuous function on $\mathbb{R}^n$ is $x \mapsto \frac{1}{2}\|x\|^2$.

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?

An excellent survey is Lagarias's [7]. In [8], 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 $\operatorname{dom} g \subseteq (\ln(2/3), \ln(2))$, 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 very 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. Due to its length, we decide not to 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`.

```
> f5 := convert(ln((1/2)*(2^x+(2/3)^x)),cf);
```

$$f5 := \left\{ \ln(\frac{1}{2}\,2^x + \frac{1}{2}\,(\frac{2}{3})^x) \quad (-\infty < x) \textbf{ and } (x < \infty) \right.$$

```
> g5 := conj(%,y): # conjugate is very long
> cfeval(g5,y=1/2*ln(4/3));
```

$$0$$

```
> temp := op([1,3,2],g5):
> ybar := fsolve(temp=y,y);
```

$$ybar := .02399367662$$

6. A barrier function

In [11], Polyak and Teboulle study algorithms for convex optimizations and they use (implicitly) convex functions and their Fenchel conjugates. (We should point out that the authors decided to consider *concave* functions and a corresponding *concave conjugates*. However, this is mathematically equivalent to our situation.) Here we recover the last example in their paper; also, we use `factor` to simplify the Fenchel conjugate. (The use of `expand` and `normal` is similar.)

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

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

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

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

7. An example from Bertsekas's text

The following pair can be found on page 468 in Bertsekas's recent text [3]. In fact, `fenchel` is good enough to recover all conjugate pairs in this book!

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

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

```
> g7 := factor(conj(f7,y));
```

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

8. An example from Rockafellar's text

The next function can be found on [13, page 229]. We must enter it via `cf`, since `piecewise` cannot properly handly two occurences of ∞. (See our comments in the Section "Why `fenchel`".) We use `cfplot` to plot the function (see Figure 1). To find its subdifferential, we use `subdiff`. Next, we plot it using `sdplot` (see Figure 2). We then find the conjugate and plot it, too (see Figure 3). Finally, we convert the conjugate to a `piecewise` function.

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

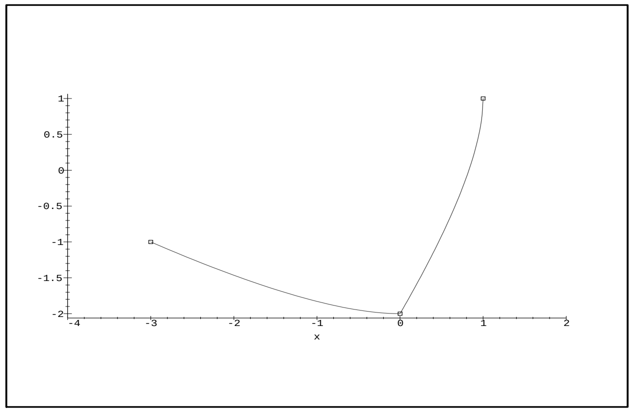

Figure 1: The graph of $f8$

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

```
> cfplot(f8,x=-4..2,scaling=constrained,
>        axes=framed);
> sdf8 := subdiff(f8);
```

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

```
> sdplot(sdf8,x=-3..1,view=[-3..1,-3..5],
>        axes=none);
> g8 := conj(f8,y);
```

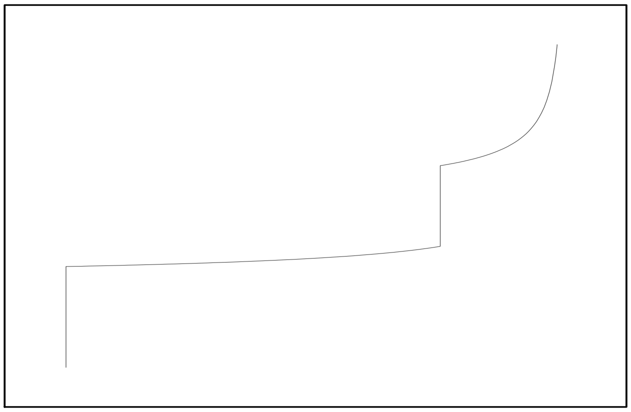

Figure 2: The subdifferential of $f8$

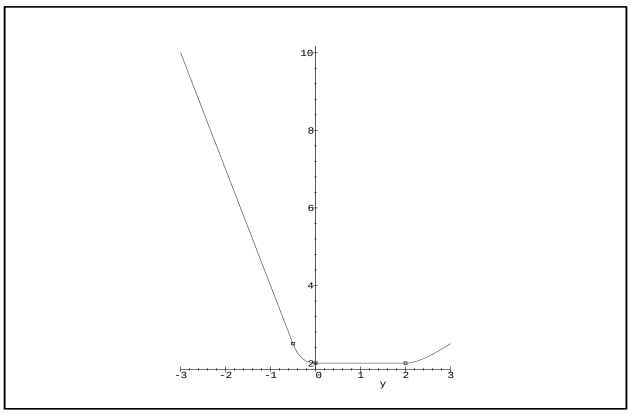

Figure 3: Graph of $g8 = (f8)^*$

$$g8 := \begin{cases} -3\,y+1 & (-\infty < y) \text{ and } (y < \frac{-1}{2}) \\ \frac{5}{2} & y = \frac{-1}{2} \\ \frac{y^2+2\,y+2}{y+1} & (\frac{-1}{2} < y) \text{ and } (y < 0) \\ 2 & y = 0 \\ 2 & (0 < y) \text{ and } (y < 2) \\ 2 & y = 2 \\ \frac{y^2-2\,y+2}{y-1} & (2 < y) \text{ and } (y < \infty) \end{cases}$$

```
> cfplot(g8,y=-3..3,scaling=constrained);
> simplify(convpw(g8));
```

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

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 $\frac{1}{2}x^2$. We also make use of `simplify`.

```
> f1;
> f9 :=  convert(1/2*x^2,cf);
```

$$\begin{cases} -x & (-\infty < x) \text{ and } (x < 0) \\ 0 & x = 0 \\ x & (0 < x) \text{ and } (x < \infty) \end{cases}$$

$$f9 := \begin{cases} \frac{1}{2}x^2 & (-\infty < x) \text{ and } (x < \infty) \end{cases}$$

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

$$\begin{cases} \infty & (-\infty < y) \text{ and } (y < -1) \\ \frac{1}{2} & y = -1 \\ \frac{1}{2}y^2 & (-1 < y) \text{ and } (y < 1) \\ \frac{1}{2} & y = 1 \\ \infty & (1 < y) \text{ and } (y < \infty) \end{cases}$$

```
> conj(%,x);
```

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

10. A case of Young's inequality

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

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

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. In our last example, we derive the conjugate pair for $p = 3$.

```
> f10 := convert(1/3*abs(x)^3,cf);
```

$$f10 := \begin{cases} -\frac{1}{3}x^3 & (-\infty < x) \textbf{ and } (x < 0) \\ 0 & x = 0 \\ \frac{1}{3}x^3 & (0 < x) \textbf{ and } (x < \infty) \end{cases}$$

```
> g10 := conj(f10,y);
```

$$g10 := \begin{cases} -\frac{2}{3}y\sqrt{-y} & (-\infty < y) \textbf{ and } (y < 0) \\ 0 & y = 0 \\ \frac{2}{3}y^{3/2} & (0 < y) \textbf{ and } (y < \infty) \end{cases}$$

Limitations

Because `limit` and `diff` are well implemented in Maple, finding subdifferentials with `subdiff` does not pose a problem. On the other hand, computing Fenchel conjugates is hard. Its key step is the inversion of the subdifferential, which relies on `solve`. It is very hard to compute some of these solutions symbolically.

If `conj` fails, it is usually due to a *problem with inverting the Fenchel/Young equality*. 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 Example 10.

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 analysis and optimization;
- `fenchel` will spark further activities in this area resulting perhaps in code that tackles nonseparable multidimensional convex functions;
- last but not least, we have been able to transmit some of our enthusiams for convex analysis to the reader.

Acknowledgment

We wish to thank Jon Borwein, Greg Fee, Serge Kruk, and Michael Monagan for helpful discussions and comments.

Biographies

Heinz H. Bauschke received his PhD in Mathematics from Simon Fraser University in 1996, after which he spent two years as an NSERC postdoctoral fellow in Waterloo (Ontario), Santa Barbara (California), and State College (Pennsylvania). He is currently a College Professor at the Okanagan University College.

His mathematical interests include convexity, optimization, functional analysis, medical imaging, and symbolic computation.

Martin von Mohrenschildt received the Doctorate in mathematics from the ETH-Zürich in Switzerland in 1994. He spent two years 94-96 as a postdoctoral fellow with the symbolic computation group of the University of Waterloo where he implemented, among other things, the new piecewise function, a pattern matcher, and a rule-based programming system.

Since August 1996, he is an Assistant Professor with the Department of Computing And Software, Faculty of Engineering, at McMaster University. He is appointed the Junior Chair in software engineering. His research interests include symbolic computation, formal software verification, and algebraic methods in software documentation.

References

[1] 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.

[2] H.H. BAUSCHKE and A.S. LEWIS. Dykstra's algorithm with Bregman projections: a convergence proof. *Optimization*. To appear.

[3] D.P. BERTSEKAS. *Nonlinear Programming*. Athena Scientific, 1995.

[4] 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.

[5] J.-B. HIRIART-URRUTY and C. LEMARÉCHAL. *Convex Analysis and Minimization Algorithms I*, volume 305 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1993.

[6] J.-B. HIRIART-URRUTY and C. LEMARÉCHAL. *Convex Analysis and Minimization Algorithms II*, volume 306 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1993.

[7] J.C. LAGARIAS. The $3x + 1$ problem and its generalizations. *American Mathematical Monthly*, 92:3–23, 1985. See also `www.cecm.sfu.ca/organics/papers`.

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

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

[10] D.S. MITRINOVIĆ, J.E. PEČARIĆ, and A.M. FINK. *Classical and New Inequalities in Analysis*, volume 61 of *Mathematics and Its Applications (East European Series)*. Kluwer, 1993.

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

[12] A.W. ROBERTS and D.E. VARBERG. *Convex Functions*. Academic Press, New York, 1973.

[13] R.T. ROCKAFELLAR. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

[14] M. von MOHRENSCHILDT. Using “piecewise” to solve classes of control theory problems. *MapleTech*, 4(3):33–37, 1997.

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