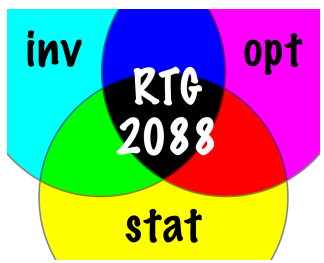
Symbolic Convex Analysis

Matthew K. Tam
Based on joint work with Florian Lauster and D. Russell Luke

Institut für Numerische und Angewandte Mathematik
Georg-August-Universität Göttingen



CARMA Seminar
May 11, 2017

Computer Algebra Systems

The focus of this talk is the manipulation of objects from convex analysis within a Computer Algebra System (CAS). Any CAS will do, but I prefer

1. 1960s: Beginning of computer algebra (*Schoonschip* by Veltman).
2. Today symbolic differentiation/integration are well-developed.
3. Why symbolic computation?
   - Removes error prone steps in tedious (but elementary) calculations.
   - Facilitates discovery of useful machinery not known to the user.
   - Makes tools easily accessible to a non-specialist or student (*e.g.*, *Wolfram Alpha*).

Computer Algebra Systems

Mathematical theory underpins good symbolic computation approaches.

1. It is always possible to work in ad-hoc manner, however an approach which systematic and theoretically justified is needed to develop robust and reliable software.
2. When working symbolically, computational complexity is often bad but the computations are rigorous and exact (when done properly).
3. Not a replacement: Fall-back to numerical approaches as needed.

Relatively little work has been done in the way of utlising symbolic computation in the convex analysis world. This will be our focus today.

Convex Analysis in 30 Seconds

Three-second overview: Study of real extended-valued, convex but not necessarily differentiable functions (*e.g.*, $f(x) = |x|$ or $\|x\|_1$).

Useful for devising algorithms to solve minimisation problems:

$$\inf_{x \in \mathbb{R}^n} \{f(x) + g(Ax)\}. \tag{1}$$

Constrained optimisation is just a special case:

$$\inf_{x \in C} f(x) = \inf_{x \in \mathbb{R}^n} \{f(x) + \iota_C(x)\},$$

where ι_C denotes the indicator function of the set $C \subseteq \mathbb{R}^n$ defined by

$$\iota_C(x) = \begin{cases} 0 & x \in C \\ +\infty & x \notin C. \end{cases}$$

Convex Analysis in 30 Seconds

Three-second overview: Study of real extended-valued, convex but not necessarily differentiable functions (*e.g.*, $f(x) = |x|$ or $\|x\|_1$).

Useful for devising algorithms to solve minimisation problems:

$$\inf_{x \in \mathbb{R}^n} \{f(x) + g(Ax)\}. \tag{1}$$

Constrained optimisation is just a special case:

$$\inf_{x \in C} f(x) = \inf_{x \in \mathbb{R}^n} \{f(x) + \iota_C(x)\},$$

where ι_C denotes the indicator function of the set $C \subseteq \mathbb{R}^n$ defined by

$$\iota_C(x) = \begin{cases} 0 & x \in C \\ +\infty & x \notin C. \end{cases}$$

Convex Analysis in 30 Seconds

Three-second overview: Study of real extended-valued, convex but not necessarily differentiable functions (*e.g.*, $f(x) = |x|$ or $\|x\|_1$).

Useful for devising algorithms to solve minimisation problems:

$$\inf_{x \in \mathbb{R}^n} \{f(x) + g(Ax)\}. \tag{1}$$

Constrained optimisation is just a special case:

$$\inf_{x \in C} f(x) = \inf_{x \in \mathbb{R}^n} \{f(x) + \iota_C(x)\},$$

where ι_C denotes the indicator function of the set $C \subseteq \mathbb{R}^n$ defined by

$$\iota_C(x) = \begin{cases} 0 & x \in C \\ +\infty & x \notin C. \end{cases}$$

Convex Analysis in 30 Seconds

$$p := \inf_{x \in \mathbb{R}^n} \{f(x) + g(Ax)\}. \tag{1}$$

Two results of fundamental importance are:

1. First-order optimality: If f, g are convex, then under mild conditions

$$x \in \mathbb{R}^n \text{ solves (1)} \iff 0 \in \partial f(x) + \partial g(x),$$

where "∂f" is the so-called subdifferential of a function f.

2. Fenchal duality: The dual to (1) is maximisation problem

$$d := \sup_{y \in \mathbb{R}^m} \{-f^*(A^\mathsf{T} y) - g^*(-y)\},$$

where f^* is the Fenchel conjugate of f. Then $p \geq d$ and, moreover, equality holds when a constraint qualification is satisfied.

Convex Analysis in 30 Seconds

$$p := \inf_{x \in \mathbb{R}^n} \{f(x) + g(Ax)\}. \tag{1}$$

Two results of fundamental importance are:

1. First-order optimality: If f, g are convex, then under mild conditions

$$x \in \mathbb{R}^n \text{ solves (1)} \iff 0 \in \partial f(x) + \partial g(x),$$

where "∂f" is the so-called subdifferential of a function f.

2. Fenchal duality: The dual to (1) is maximisation problem

$$d := \sup_{y \in \mathbb{R}^m} \{-f^*(A^\intercal y) - g^*(-y)\},$$

where f^* is the Fenchel conjugate of f. Then $p \geq d$ and, moreover, equality holds when a constraint qualification is satisfied.

Convex Analysis on a Computer?

Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be an extended real-valued function. Two of the workhorses of convex analysis are:

- The Fenchel conjugate of f given by

$$f^*(y) := \sup_{x \in \mathbb{R}^n} \{\langle x, y \rangle - f(x)\rangle\}.$$

- The subdifferential of f given by

$$\partial f(\overline{x}) := \{\phi \in \mathbb{R}^n : \langle \phi, x - \overline{x} \rangle \leq f(x) - f(\overline{x})\},$$

whenever $\overline{x} \in \operatorname{dom} f$, and equal to $\emptyset$ otherwise.

How can we *compute* these objects? What does *compute* mean?

Convex Analysis on a Computer?

Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be an extended real-valued function. Two of the workhorses of convex analysis are:

- The Fenchel conjugate of f given by

$$f^*(y) := \sup_{x \in \mathbb{R}^n} \{\langle x, y \rangle - f(x)\rangle\}.$$

- The subdifferential of f given by

$$\partial f(\overline{x}) := \{\phi \in \mathbb{R}^n : \langle \phi, x - \overline{x} \rangle \leq f(x) - f(\overline{x})\},$$

whenever $\overline{x} \in \operatorname{dom} f$, and equal to $\emptyset$ otherwise.

How can we *compute* these objects? What does *compute* mean?

Numerical Fenchel Conjugatation

Let $f : \mathbb{R} \to (-\infty, +\infty]$ be proper, lsc and convex. One approach to numerically compute the conjugate function follows. Recall that

$$f^*(y) := \sup\{\langle x, y\rangle - f(x) : x \in \mathbb{R}\}.$$

Choose sets $X := \{x_1, \ldots, x_n\} \subseteq \mathbb{R}$ and $Y := \{y_1, \ldots, y_m\} \subseteq \mathbb{R}$. Then

$$\begin{aligned} f^*(y_i) &= \sup\{\langle x, y_i\rangle - f(x) : x \in X\} \\ &\approx \sup\{\langle x_j, y_i\rangle - f(x_j) : j = 1, \ldots, n\}. \end{aligned}$$

What is (theoretically) possible with this approach:

- Approximation converges pointwise to f^* as $n, m \to +\infty$.
- Brute force complexity is $\mathcal{O}(nm)$.
- FFT-type approaches have complexity $\mathcal{O}((n+m)\log(n+m))$.
- Lucet's *Linear-time Lengendre Transform* has complexity $\mathcal{O}(n+m)$.

Numerical Fenchel Conjugatation

Let $f : \mathbb{R} \to (-\infty, +\infty]$ be proper, lsc and convex. One approach to numerically compute the conjugate function follows. Recall that

$$f^*(y) := \sup\{\langle x, y\rangle - f(x) : x \in \mathbb{R}\}.$$

Choose sets $X := \{x_1, \ldots, x_n\} \subseteq \mathbb{R}$ and $Y := \{y_1, \ldots, y_m\} \subseteq \mathbb{R}$. Then

$$\begin{aligned} f^*(y_i) &= \sup\{\langle x, y_i\rangle - f(x) : x \in X\} \\ &\approx \sup\{\langle x_j, y_i\rangle - f(x_j) : j = 1, \ldots, n\}. \end{aligned}$$

What is (theoretically) possible with this approach:

- Approximation converges pointwise to f^* as $n, m \to +\infty$.
- Brute force complexity is $\mathcal{O}(nm)$.
- FFT-type approaches have complexity $\mathcal{O}((n+m)\log(n+m))$.
- Lucet's *Linear-time Lengendre Transform* has complexity $\mathcal{O}(n+m)$.

Numerical Fenchel Conjugatation

Let $f : \mathbb{R} \to (-\infty, +\infty]$ be proper, lsc and convex. One approach to numerically compute the conjugate function follows. Recall that

$$f^*(y) := \sup\{\langle x, y\rangle - f(x) : x \in \mathbb{R}\}.$$

Choose sets $X := \{x_1, \ldots, x_n\} \subseteq \mathbb{R}$ and $Y := \{y_1, \ldots, y_m\} \subseteq \mathbb{R}$. Then

$$\begin{aligned} f^*(y_i) &= \sup\{\langle x, y_i\rangle - f(x) : x \in X\} \\ &\approx \sup\{\langle x_j, y_i\rangle - f(x_j) : j = 1, \ldots, n\}. \end{aligned}$$

What is (theoretically) possible with this approach:

- Approximation converges pointwise to f^* as $n, m \to +\infty$.
- Brute force complexity is $\mathcal{O}(nm)$.
- FFT-type approaches have complexity $\mathcal{O}((n+m)\log(n+m))$.
- Lucet's *Linear-time Lengendre Transform* has complexity $\mathcal{O}(n+m)$.

Symbolic Computation: A Quick Primer

Consider the function

$$f(x) := x \log(x).$$

Here is an example of some basic symbolic operators (in Maple):

- Define a function: `f:= x->x*log(x);`
- Take a limit (even if f not defined): `limit(f(x),x=0);`
- Differentiate an expression: `diff(f(x),x);`
- Solve equations: `solve(f(x)=y,x);`

Pause to consider how these might be implemented – it's non-trival!

Just to illustrate that much more is possible, here's another example:

```
BI:=n->Int(Product(sin(x/(2*k+1))/(x/(2*k+1)),k=0..n),x=0..infinity);
for n from 0 to 7 do
    BI(n) = value(BI(n));
end do;
```

Symbolic Computation: A Quick Primer

Consider the function

$$f(x) := x \log(x).$$

Here is an example of some basic symbolic operators (in Maple):

- Define a function: `f:= x->x*log(x);`
- Take a limit (even if f not defined): `limit(f(x),x=0);`
- Differentiate an expression: `diff(f(x),x);`
- Solve equations: `solve(f(x)=y,x);`

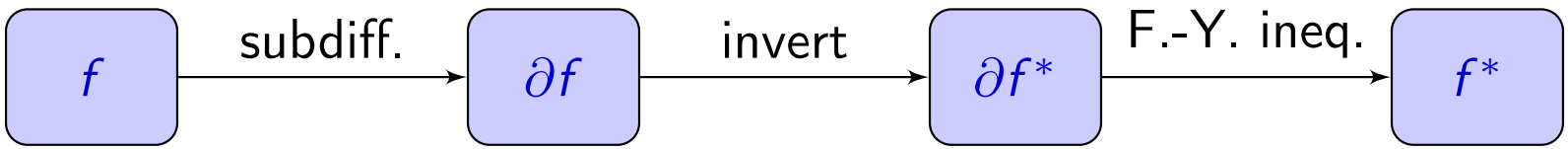
Pause to consider how these might be implemented – it's non-trival!

Just to illustrate that much more is possible, here's another example:

```
BI:=n->Int(Product(sin(x/(2*k+1))/(x/(2*k+1)),k=0..n),x=0..infinity);
for n from 0 to 7 do
    BI(n) = value(BI(n));
end do;
```

The Symbolic Approach to Fenchel Conjugation

Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be proper, lsc and convex. The basic idea for symbolic computation of the Fenchel conjugate is as follows:



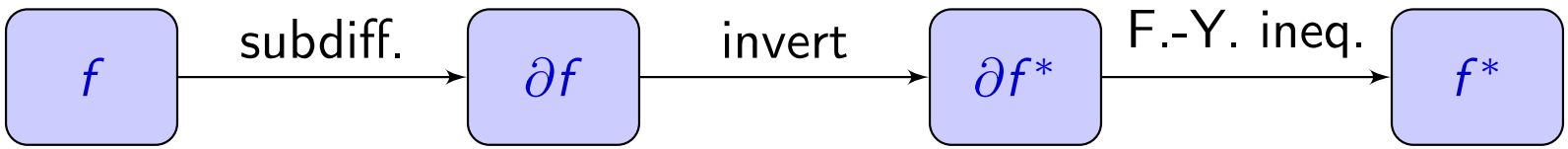
A consequence of the Fenchel–Young inequality is that

$$f^*(y) = \langle x, y \rangle - f(x) \text{ whenever } x \in \partial f^*(y).$$

- Inversion of ∂f can be problematic (Think quintic equations).
- Data-structures which represents f must be able to represent f^*.
- The same representativity requirement applies to ∂f and ∂f^*.

The Symbolic Approach to Fenchel Conjugation

Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be proper, lsc and convex. The basic idea for symbolic computation of the Fenchel conjugate is as follows:



A consequence of the Fenchel–Young inequality is that

$$f^*(y) = \langle x, y \rangle - f(x) \text{ whenever } x \in \partial f^*(y).$$

- Inversion of ∂f can be problematic (Think quintic equations).
- Data-structures which represents f must be able to represent f^*.
- The same representativity requirement applies to ∂f and ∂f^*.

A Suitable Family of Convex Functions

Definition ($\mathcal{F}$-functions)

For a set of finitely many points $A = \{a_i\}_{i=1}^m$ satisfying

$$a_0 = -\infty < a_1 < \cdots < a_{m-1} < a_m = +\infty, \tag{2}$$

we say a function $f : \mathbb{R} \to (-\infty, +\infty]$ belongs to $\mathcal{F}(A)$ if:

(a) f is lsc and convex;

(b) f is continuous on its effective domain; and

(c) the restriction of f to (a_i, a_{i+1}) is either:

- (i) affine,
- (ii) differentiable and strictly convex, or
- (iii) identically equal to $+\infty$.

The class $\mathcal{F}$ is the union of $\mathcal{F}(A)$ over all finite sets A satisfying (2)

Studied by Bauschke & von Mohrenschildt, and Borwein & Hamilton.

The "Infamous" Absolute Value

Consider the function $f(x) = |x|$ which is clearly contained in $\mathcal{F}$.

Let's take a look at how we compute with this symbolically. Write:

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

We have:

- $a_0 = -\infty, a_1 = 0$ and $a_2 = +\infty$.
- f is affine on (a_0, a_1) and (a_1, a_2).

Let's compute its subdifferential.

The "Infamous" Absolute Value

Consider the function $f(x) = |x|$ which is clearly contained in $\mathcal{F}$.

Let's take a look at how we compute with this symbolically. Write:

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

We have:

- $a_0 = -\infty, a_1 = 0$ and $a_2 = +\infty$.
- f is affine on (a_0, a_1) and (a_1, a_2).

Let's compute its subdifferential.

An entropy example

Consider the function $g(x) = x \log x$ on $\mathbb{R}_+$ with $g(0) := 0$.

Let's take a look at how we compute with this symbolically. Write:

$$g(x) = \begin{cases} x \log x & x > 0, \\ 0 & x = 0, \\ +\infty & x < 0. \end{cases}$$

We have:

- $a_0 = -\infty, a_1 = 0$ and $a_2 = +\infty$.
- g is differentiable and strictly convex on (a_1, a_2).
- (a_0, a_1) is outside the domain of g.

Let's compute its subdifferential.

An entropy example

Consider the function $g(x) = x \log x$ on $\mathbb{R}_+$ with $g(0) := 0$.

Let's take a look at how we compute with this symbolically. Write:

$$g(x) = \begin{cases} x \log x & x > 0, \\ 0 & x = 0, \\ +\infty & x < 0. \end{cases}$$

We have:

- $a_0 = -\infty, a_1 = 0$ and $a_2 = +\infty$.
- g is differentiable and strictly convex on (a_1, a_2).
- (a_0, a_1) is outside the domain of g.

Let's compute its subdifferential.

A nonstandard proof of convexity

It is worth keeping in mind that the 'obvious' human approach is not always the best symbolic approach. For instance, the following fact gives a pathway to proving convexity.

Fact (Fenchel biconjugation)

For any function $f : \mathbb{R}^n \to (-\infty, +\infty]$, we have

$$f \text{ is lsc and convex} \iff f = f^{**}.$$

In fact, the operation of Fenchel conjugation induces an order-inverting bijection between proper lsc convex functions.

Claim. The functions $f = |\cdot|$ and $g = x \log x$ are proper, lsc and convex.

Proof. Using Maple, check that the biconjugates equal the function:

```
f-Conj(Conj(f,y),x);
g-Conj(Conj(g,y),x);
```

Monotone Operators

Let $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be a set-valued map.

- T is monotone if

$$\left.\begin{array}{l}(x, x^+) \in \operatorname{gph} T \\ (y, y^+) \in \operatorname{gph} T\end{array}\right\} \implies \langle x - y, x^+ - y^+ \rangle \geq 0.$$

- T is cyclical monotone if, for every $n \geq 2$, it holds that

$$\left.\begin{array}{c}(x_1, x_1^+) \in \operatorname{gph} T \\ \vdots \\ (x_n, x_n^+) \in \operatorname{gph} T \\ x_{n+1} = x_1\end{array}\right\} \implies \sum_{i=1}^{n} \langle x_{i+1} - x_i, x_i^+ \rangle \geq 0.$$

- T is maximal (cyclic) monotone if:

$$\bar{T} \text{ (cyclic) monotone and } \operatorname{gph} T \subseteq \operatorname{gph} \bar{T} \text{ implies } T = \bar{T}.$$

- If f is proper, lsc and convex then ∂f is maximal cyclic monotone.

Monotone Operators

Let $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be a set-valued map.

- T is monotone if

$$\left.\begin{array}{l}(x, x^+) \in \operatorname{gph} T \\ (y, y^+) \in \operatorname{gph} T\end{array}\right\} \implies \langle x - y, x^+ - y^+ \rangle \geq 0.$$

- T is cyclical monotone if, for every $n \geq 2$, it holds that

$$\left.\begin{array}{c}(x_1, x_1^+) \in \operatorname{gph} T \\ \vdots \\ (x_n, x_n^+) \in \operatorname{gph} T \\ x_{n+1} = x_1\end{array}\right\} \implies \sum_{i=1}^{n} \langle x_{i+1} - x_i, x_i^+ \rangle \geq 0.$$

- T is maximal (cyclic) monotone if:

$$\bar{T} \text{ (cyclic) monotone and } \operatorname{gph} T \subseteq \operatorname{gph} \bar{T} \text{ implies } T = \bar{T}.$$

- If f is proper, lsc and convex then ∂f is maximal cyclic monotone.

A Suitable Family of Monotone Operators

We propose the follow family of monotone operators.

Definition ($\mathcal{T}$-operators)

For a set of finitely many points $B = \{b_i\}_{i=0}^{l}$ satisfying

$$b_0 = -\infty < b_1 < \cdots < b_{l-1} < b_l = +\infty, \tag{3}$$

we say a set-valued operator $T : \mathbb{R} \rightrightarrows \mathbb{R}$ belongs to $\mathcal{T}(B)$ if there exists a maximal monotone extension $\widetilde{T}$ of T such that the restriction $\mathcal{T}$ to each interval (b_i, b_{i+1}) is either

(i) single-valued and constant;

(ii) single-valued, continuous and strictly monotone; or

(iii) identically equal to the empty-set.

The class $\mathcal{T}$ is the union of $\mathcal{T}(B)$ over all finite sets B satisfying (3).

Important Properties of $\mathcal{T}$-operators

All the most important closure properties for the class hold.

Proposition (Properties of $\mathcal{T}$-operators)

The following assertions hold.

(a) If $T \in \mathcal{T}$ and $\lambda \geq 0$ then $\lambda T \in \mathcal{T}$.

(b) If $T_1, T_2 \in \mathcal{T}$ then $T_1 + T_2 \in \mathcal{T}$.

(c) If $T \in \mathcal{T}$ then $T^{-1} \in \mathcal{T}$.

(d) If $T \in \mathcal{T}$ and $\lambda > 0$ then $(I + \lambda T)^{-1} \in \mathcal{T}$.

(e) If $T_1, T_2 \in \mathcal{T}$ then $(T_1^{-1} + T_2^{-1})^{-1} \in \mathcal{T}$.

Property (d) states that $\mathcal{T}$-operators are closed under taking resolvents. In particular, if $T = \partial f$ then the proximity operator of f is also in $\mathcal{T}$.

Important Properties of $\mathcal{T}$-operators

All the most important closure properties for the class hold.

Proposition (Properties of $\mathcal{T}$-operators)

The following assertions hold.

(a) If $T \in \mathcal{T}$ and $\lambda \geq 0$ then $\lambda T \in \mathcal{T}$.

(b) If $T_1, T_2 \in \mathcal{T}$ then $T_1 + T_2 \in \mathcal{T}$.

(c) If $T \in \mathcal{T}$ then $T^{-1} \in \mathcal{T}$.

(d) If $T \in \mathcal{T}$ and $\lambda > 0$ then $(I + \lambda T)^{-1} \in \mathcal{T}$.

(e) If $T_1, T_2 \in \mathcal{T}$ then $(T_1^{-1} + T_2^{-1})^{-1} \in \mathcal{T}$.

Property (d) states that $\mathcal{T}$-operators are closed under taking resolvents.
In particular, if $T = \partial f$ then the proximity operator of f is also in $\mathcal{T}$.

Connections Between $\mathcal{F}$ and $\mathcal{T}$

We summarizes the connection between $\mathcal{F}$-functions and $\mathcal{T}$-operators.

Theorem ("$\mathcal{T} = \partial\mathcal{F}$")

If $f \in \mathcal{F}$ is proper then ∂f is maximal monotone and belongs to $\mathcal{T}$. Conversely, if $T \in \mathcal{T}$ is maximal monotone then there exists a proper, lsc, convex function f such that $T = \partial f$ and, moreover, any such function belongs to $\mathcal{F}$.

Theorem ("$\mathcal{F}^* = \mathcal{F}$")

If $f \in \mathcal{F}$ is a proper function then $f^* \in \mathcal{F}$.

The key step of the proof is: if $\partial f \in \mathcal{T}$ then $(\partial f)^{-1} = \partial f^* \in \mathcal{T}$.

Connections Between $\mathcal{F}$ and $\mathcal{T}$

We summarizes the connection between $\mathcal{F}$-functions and $\mathcal{T}$-operators.

Theorem ("$\mathcal{T} = \partial\mathcal{F}$")

If $f \in \mathcal{F}$ is proper then ∂f is maximal monotone and belongs to $\mathcal{T}$. Conversely, if $T \in \mathcal{T}$ is maximal monotone then there exists a proper, lsc, convex function f such that $T = \partial f$ and, moreover, any such function belongs to $\mathcal{F}$.

Theorem ("$\mathcal{F}^* = \mathcal{F}$")

If $f \in \mathcal{F}$ is a proper function then $f^* \in \mathcal{F}$.

The key step of the proof is: if $\partial f \in \mathcal{T}$ then $(\partial f)^{-1} = \partial f^* \in \mathcal{T}$.

Closure of $\mathcal{F}$ under Fenchel Conjugation

Assumption (c) of the $\mathcal{F}$-function definition is essential for closure under Fenchel conjugation. Recall:

Definition (Part (c) of $\mathcal{F}$-functions)

(c) the restriction of f to (a_i, a_{i+1}) is either:

- (i) affine,
- (ii) differentiable and strictly convex, or
- (iii) identically equal to $+\infty$.

In its absence, the closure can fail spectacularly.

Example (Lauster)

There exists a convex $f : \mathbb{R} \to \mathbb{R}$ which is infinitely differentiable on $\mathbb{R} \setminus \{0\}$ such that its Fenchel conjugate, f^*, is not differentiable at infinitely many points in its domain. In particular, $f^* \notin \mathcal{F}$.

Examples and Illustrations

I will now give three example illustrations of the framework. Namely,

1. Explicit Formula for Proximity Operators
2. Recovery of Penalty Functions
3. Superexpectations, Superdistributions and Superquantiles

The examples are programmed in `Maple` and use the `SCAT` package where appropriate. The code can be found online at:

`http://vaopt.math.uni-goettingen.de/software.php`

Or accessed through GWDG's GitLab repository:

`https://gitlab.gwdg.de/mtam/SymCA/`

Examples and Illustrations

I will now give three example illustrations of the framework. Namely,

1. Explicit Formula for Proximity Operators
2. Recovery of Penalty Functions
3. Superexpectations, Superdistributions and Superquantiles

The examples are programmed in `Maple` and use the `SCAT` package where appropriate. The code can be found online at:

`http://vaopt.math.uni-goettingen.de/software.php`

Or accessed through GWDG's GitLab repository:

`https://gitlab.gwdg.de/mtam/SymCA/`

1. Explicit Formula for Proximity Operators

The proximity operator of a function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is given by

$$\text{prox}_f := \underset{y \in \mathbb{R}^n}{\arg\min} \left\{ f(y) + \frac{1}{2} \| \cdot - y \|^2 \right\}.$$

When f is proper, lsc and convex, one has

$$\text{prox}_f = (I + \partial f)^{-1}.$$

That is, the proximity operator is the resolvent of the subdifferential.

The following approach is thus theoretically justified:

$$f \in \mathcal{F} \overset{\text{Th.}}{\Longrightarrow} \partial f \in \mathcal{T} \overset{\text{Prop.(d)}}{\Longrightarrow} (I + \partial f)^{-1} \in \mathcal{T}.$$

1. Explicit Formula for Proximity Operators

The proximity operator of a function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is given by

$$\text{prox}_f := \underset{y \in \mathbb{R}^n}{\arg\min} \left\{ f(y) + \frac{1}{2} \| \cdot - y \|^2 \right\}.$$

When f is proper, lsc and convex, one has

$$\text{prox}_f = (I + \partial f)^{-1}.$$

That is, the proximity operator is the resolvent of the subdifferential.

The following approach is thus theoretically justified:

$$f \in \mathcal{F} \overset{\text{Th.}}{\Longrightarrow} \partial f \in \mathcal{T} \overset{\text{Prop.(d)}}{\Longrightarrow} (I + \partial f)^{-1} \in \mathcal{T}.$$

1. Explicit Formula for Proximity Operators

The proximity operator of a function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is given by

$$\text{prox}_f := \underset{y \in \mathbb{R}^n}{\arg\min} \left\{ f(y) + \frac{1}{2} \| \cdot - y \|^2 \right\}.$$

When f is proper, lsc and convex, one has

$$\text{prox}_f = (I + \partial f)^{-1}.$$

That is, the proximity operator is the resolvent of the subdifferential.

The following approach is thus theoretically justified:

$$f \in \mathcal{F} \overset{\text{Th.}}{\Longrightarrow} \partial f \in \mathcal{T} \overset{\text{Prop.(d)}}{\Longrightarrow} (I + \partial f)^{-1} \in \mathcal{T}.$$

1. Explicit Formula for Proximity Operators

```
with(SCAT): # load the SCAT package
f := convert(abs(x), PWF); # define f and convert to PWF
format
```

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

```
sdf := SubDiff(f):  # compute the subdiff of f
Assume(lambda > 0):
prox['f']^lambda = Invert(simplify(x+lambda*sdf),y); # prox f
```

$$prox_f^{\lambda} = \begin{cases} \{\lambda + y\} & y < -\lambda \\ \{0\} & y = -\lambda \\ \{0\} & (-\lambda < y) \text{ and } (y < \lambda) \\ \{0\} & y = \lambda \\ \{-\lambda + y\} & \lambda < y \end{cases}$$

1. Explicit Formula for Proximity Operators

```
with(SCAT): # load the SCAT package
Assume(a < b):  # assume that the interval is proper
f := convert(piecewise(a<=x and x<=b, 0, infinity), PWF, x);
```

$$f := \begin{cases} \infty & x < a \\ 0 & x = a \\ 0 & (a < x) \text{ and } (x < b) \\ 0 & x = b \\ \infty & b < x \end{cases}$$

```
sdf := SubDiff(f):  # compute the subdiff of f
P[[a, b]] = Invert(simplify(x+sdf),y); # proj onto [a,b]
```

$$P_{[a,b]} = \begin{cases} \{a\} & x < a \\ \{a\} & x = a \\ \{y\} & (a < y) \text{ and } (y < b) \\ \{b\} & x = b \\ \{b\} & b < x \end{cases}$$

2. Recovery of Penalty Functions

Given a monotone operator $T : \mathbb{R} \rightrightarrows \mathbb{R}$, we consider the problem of finding a so-called penalty function $f : \mathbb{R} \to (-\infty, +\infty]$. That is, a function f who subdifferential can be identified with T in the sense that

$$\operatorname{gph} T \subseteq \operatorname{gph}(\operatorname{prox}_f).$$

Or equivalently, such that for all $x \in \mathbb{R}$,

$$T(x) \subseteq \operatorname{prox}_f(x).$$

The problem was studied by Bayram (2015) without utilising symbolic computational tools.

2. Recovery of Penalty Functions

Proposition (Recovery of penalty functions)

Let $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be a maximal cyclical monotone operator. There exists a proper, lsc function f such that $f + \frac{1}{2}\|\cdot\|^2$ convex and $T = \operatorname{prox}_f$. Furthermore, if $T \in \mathcal{F}$ then there is an f such that $f + \frac{1}{2}\|\cdot\|^2 \in \mathcal{F}$.

Proof sketch. Take $f = h^*(y) - \frac{1}{2}\|y\|^2$. This is justified because

$$\begin{aligned} T(x) = \partial h(x) = (\partial h^*)^{-1}(x) &= \{y \in \mathbb{R}^n : x \in \partial h^*(y)\} \\ &= \operatorname*{arg\,min}_{y \in \mathbb{R}^n} \{h^*(y) - \langle x, y\rangle\} \\ &= \operatorname*{arg\,min}_{y \in \mathbb{R}^n} \left\{ \left(h^*(y) - \frac{1}{2}\|y\|^2 \right) + \frac{1}{2}\|x - y\|^2 \right\}. \end{aligned}$$

The following approach is thus theoretically justified:

$$T \in \mathcal{T} \overset{\text{integ.}}{\Longrightarrow} h \in \mathcal{F} \overset{\text{Prop.}}{\Longrightarrow} h^* \in \mathcal{F} \overset{-\frac{1}{2}\|\cdot\|^2}{\Longrightarrow} \text{penalty function}$$

2. Recovery of Penalty Functions

Proposition (Recovery of penalty functions)

Let $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be a maximal cyclical monotone operator. There exists a proper, lsc function f such that $f + \frac{1}{2}\|\cdot\|^2$ convex and $T = \operatorname{prox}_f$. Furthermore, if $T \in \mathcal{F}$ then there is an f such that $f + \frac{1}{2}\|\cdot\|^2 \in \mathcal{F}$.

Proof sketch. Take $f = h^*(y) - \frac{1}{2}\|y\|^2$. This is justified because

$$\begin{aligned} T(x) = \partial h(x) = (\partial h^*)^{-1}(x) &= \{y \in \mathbb{R}^n : x \in \partial h^*(y)\} \\ &= \operatorname*{arg\,min}_{y \in \mathbb{R}^n} \{h^*(y) - \langle x, y\rangle\} \\ &= \operatorname*{arg\,min}_{y \in \mathbb{R}^n} \left\{ \left(h^*(y) - \frac{1}{2}\|y\|^2 \right) + \frac{1}{2}\|x - y\|^2 \right\}. \end{aligned}$$

The following approach is thus theoretically justified:

$$T \in \mathcal{T} \overset{\text{integ.}}{\Longrightarrow} h \in \mathcal{F} \overset{\text{Prop.}}{\Longrightarrow} h^* \in \mathcal{F} \overset{-\frac{1}{2}\|\cdot\|^2}{\Longrightarrow} \text{penalty function}$$

2. Recovery of Penalty Functions

```
with(SCAT): # load the SCAT package
Assume(alpha > 0):
H := SD([x, -alpha, 0, x, 0, alpha, 0, x, x], [x], x::real);
```

$$H := \begin{cases} \{x\} & x < -\alpha \\ \{0, x\} & x = -\alpha \\ \{0\} & (-\alpha < x) \text{ and } (x < \alpha) \\ \{0, x\} & x = \alpha \\ \{x\} & x < \alpha \end{cases}$$

```
Conjh := Conj(Integ(H),y):
f = factor(simplify(Conjh-1/2*y^2));
```

$$f = \begin{cases} 0 & y < -\alpha \\ \frac{1}{2}(\alpha - y)(\alpha + y) & y = -\alpha \\ \frac{1}{2}(\alpha + y)^2 & (-\alpha < y) \text{ and } (y < 0) \\ -\frac{1}{2}\alpha^2 - \frac{1}{2}y^2 & y = 0 \\ -\frac{1}{2}(\alpha - y)^2 & (0 < y) \text{ and } (y < \alpha) \\ \frac{1}{2}(\alpha - y)(\alpha + y) & y = \alpha \\ 0 & \alpha < y \end{cases}$$

3. Superexpectations, Superdistributions and Superquantiles

The following example is from Rockafellar–Royset, 2013.

Example (exponential distributions). Let X be exponentially distributed with parameter $\lambda > 0$. Then the distribution function is $F_X(x) = 1 - \exp(-\lambda x)$, the superexpectation function is

$$E_X(x) = \begin{cases} x + (1/\lambda)\exp(-\lambda x) & \text{for } x \geq 0, \\ 1/\lambda & \text{for } x < 0, \end{cases}$$

and the conjugate superexpection function is $E_X^*(p) = (1/\lambda)(p-1)(1-\log(1-p))$ for $p \in [0,1)$. Quantile and superquantiles are thus given on $(0,1)$ by

$$Q_X(p) = -(1/\lambda)\log(1-p), \qquad \overline{Q}_X(p) = (1/\lambda)[1-\log(1-p)].$$

Starting with the distribution function F_X, we can symbolic compute the various objects. In particular, the superexpection function

$$E_X(x) := \mathbb{E}(\max\{x, X\}),$$

is suited for symbolic computation not through its definition but rather by integration of F_X.

Rockafellar & Royset

Theorem 1 (characterization of superexpectations). *The superexpectation function E_X for a random variable X having $E[\,|X|\,] < \infty$ is a finite convex function on $(-\infty, \infty)$ which corresponds subdifferentially to the monotone relation Γ_X and the distribution function F_X through*

$$\Gamma_X = \operatorname{gph} \partial E_X, \qquad F_X(x) = E'^{+}_X(x). \tag{3.14}$$

It is nondecreasing with

$$E_X(x) - x \geq 0, \qquad \lim_{x\to\infty}[E_X(x) - x] = 0, \qquad \lim_{x\to-\infty} E_X(x) = E[X] \tag{3.15}$$

and has the additional convexity property that

$$E_X(x) \leq (1-\lambda)E_{X_0}(x) + \lambda E_{X_1}(x) \quad \text{when } X = (1-\lambda)X_0 + \lambda X_1 \text{ with } 0 < \lambda < 1. \tag{3.16}$$

On the other hand, any convex function f on $(-\infty, \infty)$ with the properties that

$$f(x) - x \geq 0, \qquad \lim_{x\nearrow\infty}[f(x) - x] = 0, \qquad \lim_{x\searrow-\infty} f(x) = \text{a finite value}, \tag{3.17}$$

is E_X for a random variable X having $E[\,|X|\,] < \infty$.

Rockafellar & Royset

Theorem 1 (characterization of superexpectations). *The superexpectation function E_X for a random variable X having $E[\,|X|\,] < \infty$ is a finite convex function on $(-\infty, \infty)$ which corresponds subdifferentially to the monotone relation Γ_X and the distribution function F_X through*

$$\Gamma_X = \operatorname{gph} \partial E_X, \qquad F_X(x) = E'^{+}_X(x). \tag{3.14}$$

It is nondecreasing with

$$E_X(x) - x \geq 0, \qquad \lim_{x\to\infty}[E_X(x) - x] = 0, \qquad \lim_{x\to-\infty} E_X(x) = E[X] \tag{3.15}$$

and has the additional convexity property that

$$E_X(x) \leq (1-\lambda)E_{X_0}(x) + \lambda E_{X_1}(x) \quad \text{when } X = (1-\lambda)X_0 + \lambda X_1 \text{ with } 0 < \lambda < 1. \tag{3.16}$$

On the other hand, any convex function f on $(-\infty, \infty)$ with the properties that

$$f(x) - x \geq 0, \qquad \lim_{x\nearrow\infty}[f(x) - x] = 0, \qquad \lim_{x\searrow-\infty} f(x) = \text{a finite value}, \tag{3.17}$$

is E_X for a random variable X having $E[\,|X|\,] < \infty$.

3. Rockafellar & Royset (cont.)

```
with(SCAT):
F := 1-exp(-lambda*x):  # distribution fn of X
Q := solve(F=p,x); # quantile fn of X
```

$$Q := -\frac{\ln(1-p)}{\lambda}$$

```
# superquantile fn of X
superQ := factor(1/(1-p)*int(subs(p=t,Q),t=p..1));
```

$$superQ := -\frac{\ln(1-p)-1}{\lambda}$$

```
Assume(lambda>0):
F := convert(piecewise(x >= 0,F), SD,x);
```

$$E := \begin{cases} \{0\} & x < 0 \\ \{0\} & x = 0 \\ \{1-e^{-\lambda x}\} & 0 < x \end{cases}$$

3. Rockafellar & Royset (cont.)

```
# compute the superexpection function of F
E0 := Integ(F,x):
c0 := Eval(simplify(E0-x), x = infinity):
E := simplify(E0 - c0);
```

$$E := \begin{cases} \frac{1}{\lambda} & x < 0 \\ \frac{1}{\lambda} & x = 0 \\ \frac{\lambda x + e^{-\lambda x}}{\lambda} & 0 < x \end{cases}$$

```
conjE := conjE(E,p,x); # the conjugate of the superexpectation
```

$$conjE := \begin{cases} \infty & p < 0 \\ -\frac{1}{\lambda} & p = 0 \\ -\frac{(-1+p)(\ln(1-p)-1)}{\lambda} & (0 < p) \text{ and } (p < 1) \\ 0 & p = 1 \\ \infty & 1 < p \end{cases}$$

Conclusions and Outlook

- The classes of $\mathcal{F}$-functions and $\mathcal{T}$-operators are well-suited to manipulation within a CAS and encompasses important examples.
- Robustness is a consequence of theoretical underpinnings.
- Easily accessible to non-specialists and students.
- Further work: Extensions to higher dimensions via recursion.
  e.g., Define $T^1 := \mathcal{T}$ and say $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a $\mathcal{T}^n$-operator if

$$(x_2, x_3, \ldots, x_n) \mapsto T(t, x_2, x_3, \ldots, x_n)$$

  is a $\mathcal{T}^{n-1}$-operator, for each fixed $t \in \mathbb{R}$.


F. Lauster, D.R. Luke and M.K. Tam: Symbolic computation with monotone operators, arXiv:1703.05946 (March 2017).

Maple worksheet for examples given in this talk online at:
`http://vaopt.math.uni-goettingen.de`