Computational Optimization and Applications manuscript No.
(will be inserted by the editor)

Visualization of the ε-Subdifferential of Piecewise Linear-Quadratic Functions

Anuj Bajaj · **Warren Hare** · **Yves Lucet**

Received: date / Accepted: date

Abstract Computing explicitly the ε-subdifferential of a proper function amounts to computing the level set of a convex function namely the conjugate minus a linear function. The resulting theoretical algorithm is applied to the the class of (convex univariate) piecewise linear-quadratic functions for which existing numerical libraries allow practical computations. We visualize the results in a primal, dual, and subdifferential views through several numerical examples. We also provide a visualization of the Brøndsted-Rockafellar Theorem.

Keywords Subdifferentials · ε-Subdifferentials · Computational convex analysis (CCA) · Piecewise linear-quadratic functions · convex function · visualization

1 Introduction

Subdifferentials generalize the derivatives to nonsmooth functions, which makes them one of the most useful instruments in nonsmooth optimization. The ε-subdifferentials, which are a certain relaxation of true subdifferentials, arise naturally in cutting-plane and bundle algorithms and help overcome some limitations of subdifferential calculus of convex functions. As such, they are a useful tool in convex analysis.

We begin by defining the central concept of this work, the ε-subdifferential of a convex function.

Definition 1.1 Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, +\infty\}$ be convex, $x \in \mathrm{dom}(f)$ and $\varepsilon \geq 0$. The *ε-subdifferential* of f at x is the set

$$\partial_\varepsilon f(x) = \{s \in \mathbb{R}^n : f(y) \geq f(x) + \langle s, y - x\rangle - \varepsilon \quad \text{for all} \quad y \in \mathbb{R}^n\}.$$

A. Bajaj
Mathematics, 1132 Faculty/Administration Building, Wayne State University, 42 W. Warren Ave., Detroit, MI 48202, USA
E-mail: anuj.bajaj@alumni.ubc.ca

W. Hare
Mathematics, ASC 353, The University of British Columbia - Okanagan (UBCO), 3187 University Way, Kelowna, BC, V1V 1V7, Canada
E-mail: warren.hare@ubc.ca

Y. Lucet
Computer Science, ASC 350, UBCO
E-mail: yves.lucet@ubc.ca

The elements of $\partial_\varepsilon f(x)$ are known as the ε-*subgradients* of f at x. We define $\partial_\varepsilon f(x) = \emptyset$ when $x \notin \text{dom}(f)$. (Historically, [4] used the term "approximate subgradients", but we adopt the more common terminology of ε-subgradient to make the distinction with the approximate subdifferential introduced in [16].)

In Figure 1.1, we visualize the ε-subdifferential of a convex function for $\varepsilon = 2$ and $\varepsilon = 0$. The ε-subdifferential is the set of all vectors that create linearizations passing through $f(x) - \varepsilon$ that remain under f.

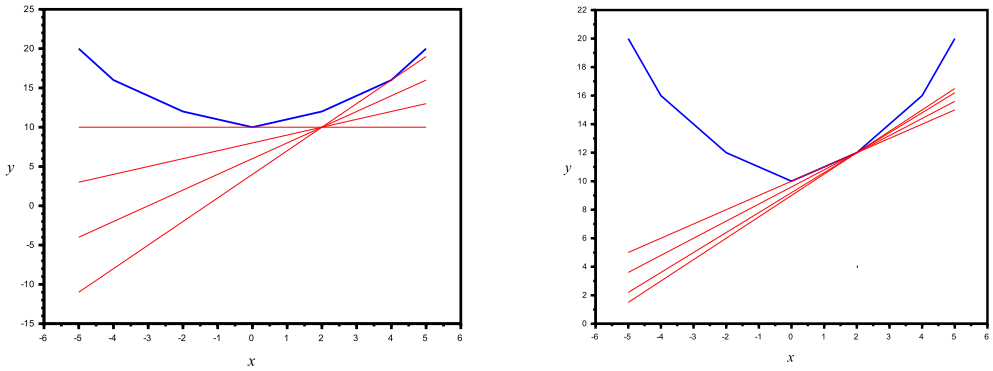


Fig. 1.1: An illustration of the construction of the ε-subdifferential of an example convex function at $x = 2$ for $\varepsilon = 2$ (left) and $\varepsilon = 0$ (right).

Notice that, for $\varepsilon = 0$ we obtain the classical subdifferential of a convex function

$$\partial f(x) = \{s \in \mathbb{R}^n : f(y) \geq f(x) + \langle s, y - x \rangle \quad \text{for all} \quad y \in \mathbb{R}^n\}.$$

It immediately follows from the definition that $\partial f(x) \subseteq \partial_\varepsilon f(x)$ for any $\varepsilon \geq 0$. Thus, the ε-subdifferential can be regarded as an enlargement of the true subdifferential.

In the context of nonsmooth optimization, various numerical methods have been developed based on the notion of subdifferentials. One group of foundational methods of particular interest to this work are *Cutting Planes* methods. Cutting Planes methods work by approximating the objective function by a piecewise linear model based on function values and subgradient vectors:

$$\check{f}_m(x) = \max_{i=1,2,\ldots,m} \{f(x^i) + \langle s^i, x - x^i \rangle\} \tag{1}$$

where $s^i \in \partial f(x^i) = \partial_0 f(x^i)$. This model is then used to guide the selection of the next iterate. Various methods based on cutting planes models exist. For example, proximal bundle methods [8, 13, 18, 20], level bundle methods [6, 23, 19], and hybrid approaches [24] (among many more).

In Figure 1.2 we illustrate 2 iterations of a very basic cutting planes method.

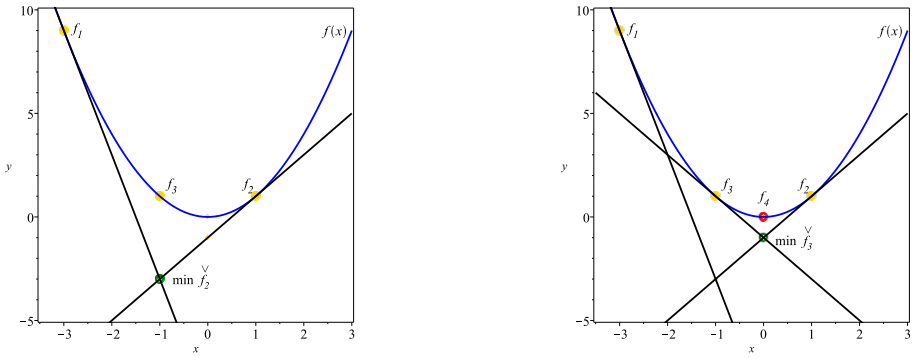


Fig. 1.2: An illustration of a simple Cutting Planes method

In Figure 1.2, we begin with points x^1 and x^2 whose function values and (sub)gradients are used to build the model $\check{f}_2$. The next iterate, x^3, is the minimizer of the model $\check{f}_2$, and the function value and a (sub)gradient at x^3 is used to refine the model and create $\check{f}_3$.

While this very basic method is generally considered ineffective [2, Example 8.1], it has lead to the plethora of methods mentioned above, and helps provide insight on how the ε-subdifferential arises naturally in nonsmooth optimization. Specifically, suppose model $\check{f}_k$ is constructed via equation (1) and used to select a new iterate x^{k+1} via the simple rule $x^{k+1} \in \arg\min \check{f}_k$. By equation (1) and the definition of the subdifferential, we have $\check{f}_k(x) \leq f(x)$ for all x. Thus,

$$\check{f}_k(x^{k+1}) \leq \check{f}_k(x) \leq f(x) \quad \text{for all} \quad x,$$

which yields

$$f(x^{k+1}) + \langle 0, x - x^{k+1} \rangle - (f(x^{k+1}) - \check{f}_k(x^{k+1})) \leq f(x) \quad \text{for all} \quad x \in \mathbb{R}. \tag{2}$$

That is,

$$0 \in \partial_{\varepsilon_k} f(x^{k+1}) \quad \text{for} \quad \varepsilon_k = f(x^{k+1}) - \check{f}_k(x^{k+1}) \geq 0.$$

This insight can lead to a proof of convergence (by proving $\varepsilon_k \to 0$) and provides stopping criterion for the algorithm. When the simple rule $x^{k+1} \in \arg\min \check{f}_k$ is replaced by more advanced methods, convergence analysis often follows a similar path, first showing $0 \in \partial_{\varepsilon_k} f(x^{k+1})$ for some appropriate choice of ε_k and then showing $\varepsilon_k \to 0$. Thus we see one example of the ε-subdifferentials role in nonsmooth optimization.

The ε-subdifferential has also been studied directly, and a number of calculus rules have been developed to help understand its behaviour [15,5]. In this work, we are interested in the development of tools to help compute and visualize the ε-subdifferential, at least in some situations. We feel that such tools will be of great value to build intuition and broader understanding of this important object in nonsmooth optimization.

In this paper, we focus on finding the ε-subdifferential of univariate convex piecewise-linear quadratic (plq) function. Such functions are of interest since they are computationally tractable [9,10,11,22,31] (see also Section 3), arise naturally as penalty functions in regularization problems [1], and arise in variety of other situations [1,7,12,25,26,28]. Moreover, any convex function can be approximated by such a convex plq function.

The present work is organized as follows. Section 2 provides some key definitions relevant to this work. Subsection 3.1 presents a general algorithm for computing the ε-subdifferential of any proper

convex function along with a few numerical examples. Subsection 3.2 presents an implementation of the general algorithm for the class of univariate convex plq functions. It also discusses the data structure and the complexity of the algorithm. Section 4 illustrates the implementation with some numerical examples, including a visualization of the classic Brøndsted-Rockafellar Theorem. Section 5 summarizes the work we have done and contains a discussion on the limitations of extending the required implementation. It also provides some directions for future work.

2 Key Definitions

In this section, we provide few key definitions required to understand this work. We assume the reader is familiar with basic definitions and results in convex analysis.

Definition 2.1 Given a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ (not necessarily convex), the *convex conjugate* (commonly known as the Fenchel Conjugate) of f denoted by f^* is defined as

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

We denote $\operatorname{dom} f = \{x \in \mathbb{R}^n : f(x) \in \mathbb{R}\}$.

Definition 2.2 A set $S \subseteq \mathbb{R}^n$ is called *polyhedral* if it can be specified as finitely many linear constraints.

$$S = \{x : \langle a_i, x\rangle \leq b_i \ , \ i = 1, 2, \ldots, p\}$$

where for $i = 1, 2, \ldots, p$, $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$.

Definition 2.3 A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, +\infty\}$ is *piecewise linear-quadratic* (plq) if $\operatorname{dom}(f)$ can be represented as the union of finitely many polyhedral sets, relative to each of which $f(x) = \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle + c$ where $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$.
Note that a plq function is continuous on its domain.

3 Algorithmic Computation of the ε-subdifferential

In this section, we propose a general algorithm that enables us to compute the ε-subdifferential for any proper function. While the algorithm would be difficult (or impossible) to implement in a general setting, we shall present an implementation specifically for univariate convex plq functions (Section 3.2). We then illustrate the implementation with some numerical examples (Section 4).

3.1 The Appx_Subdiff Algorithm

We now prove elementary results that will justify the algorithm. Note that the function m defined next is only introduced because it is already available in the CCA numerical library; it is not necessary from a theoretical viewpoint.

Proposition 3.1 *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper function, $\bar{x} \in \operatorname{dom}(f)$ and $\varepsilon > 0$. Note $l_{\bar{x}} : s \mapsto \varepsilon - f(\bar{x}) + \langle s, \bar{x}\rangle$ and $m(s) = \min\{f^*(s), l_{\bar{x}}(s)\}$. Then*

$$\partial_\varepsilon f(\bar{x}) = \{s \in \mathbb{R}^n : m(s) = f^*(s)\}. \tag{3}$$

Proof Applying the definition of m, $l_{\bar{x}}$, and f^* we obtain

$$\begin{aligned}\{s \in \mathbb{R}^n : m(s) = f^*(s)\} &= \{s \in \mathbb{R}^n : f^*(s) \leq l_{\bar{x}}(s)\},\\ &= \{s \in \mathbb{R}^n : \langle s, x\rangle - f(x) \leq \varepsilon - f(\bar{x}) + \langle s, \bar{x}\rangle,\ \forall x\},\\ &= \partial_\varepsilon f(\bar{x}).\end{aligned}$$

□

Applying Proposition 3.1 immediately produces the following algorithm for computing the ε-subdifferential of a proper function.

Algorithm 1 Appx_Subdiff Algorithm

Input: f (proper function), $\bar{x} \in \mathrm{dom}(f)$, $\varepsilon > 0$
Output: X

1: Compute $f^*(s)$
2: Define $l_{\bar{x}}(s) = \varepsilon - f(\bar{x}) + \langle s, \bar{x}\rangle$
3: Define $m(s) = \min\{f^*(s), l_{\bar{x}}(s)\}$
4: Output $X = \{s \in \mathrm{dom}(f^*) : m(s) = f^*(s)\}$

To shed some light upon the algorithm we consider the following example.

Example 3.1 Consider the function $f(x) = \dfrac{\|x\|^p}{p}$, $x \in \mathbb{R}^n$ where $1 < p < \infty$. From [3, Table 3.1] we have $f^*(s) = \dfrac{\|s\|^q}{q}$, $s \in \mathbb{R}^n$ where $\dfrac{1}{p} + \dfrac{1}{q} = 1$. We also have $l_{\bar{x}}(s) = \varepsilon - \dfrac{\|\bar{x}\|^p}{p} + \langle s, \bar{x}\rangle$. Thus, we have

$$\begin{aligned}\partial_\varepsilon f(\bar{x}) &= \{s \in \mathbb{R}^n : f^*(s) \leq l_{\bar{x}}(s)\}\\ &= \left\{s \in \mathbb{R}^n : \frac{\|s\|^q}{q} \leq \varepsilon - \frac{\|\bar{x}\|^p}{p} + \langle s, \bar{x}\rangle\right\}. \qquad (4)\end{aligned}$$

In particular, for $\bar{x} = 0$, Equation (4) becomes

$$\begin{aligned}\partial_\varepsilon f(0) &= \left\{s \in \mathbb{R}^n : \frac{\|s\|^q}{q} \leq \varepsilon\right\}\\ &= \{s \in \mathbb{R}^n : \|s\|^q \leq q\varepsilon\}\\ &= \left\{s \in \mathbb{R}^n : \|s\| \leq \left(\frac{p\varepsilon}{p-1}\right)^{(p-1)/p}\right\}.\end{aligned}$$

The particular case of $p = 5$ and $\varepsilon = 1$ is illustrated in Figure 3.1.

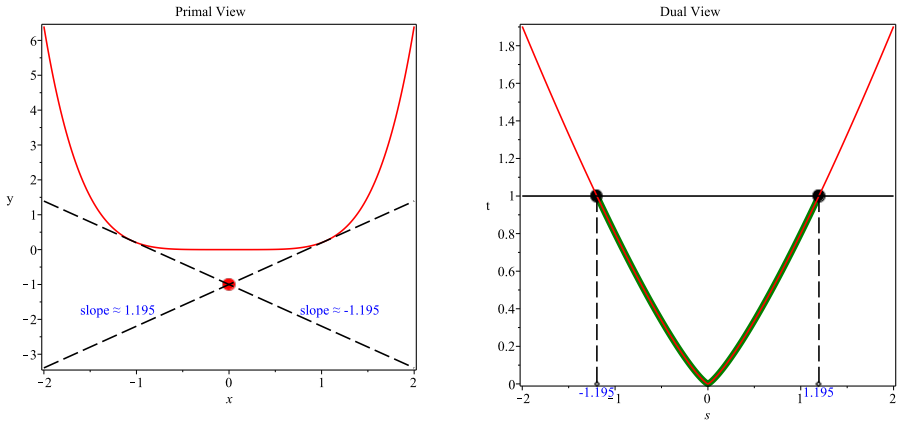


Fig. 3.1: The primal view (left) depicts the graph of $f(x) = \frac{|x|^5}{5}$ (red curve) along with the black dashed lines passing through the point $(\bar{x}, f(\bar{x}) - \varepsilon) = (0, -1)$ (red dot) having slopes -1.195 and 1.195 respectively (the lower and upper bounds of $\partial_\varepsilon f(\bar{x})$). The dual view depicts the graphs of $f^*(s) = \frac{4}{5}|s|^{5/4}$ (red curve) and $l_{\bar{x}}(s)$ (solid black line). The green curve shows when the graphs of $m(s)$ and $f^*(s)$ coincide.

Given the framework of Algorithm 1, a natural question to ask is whether there exists a collection of functions which allows for a general implementation. As mentioned, we consider the well-known class in Nonsmooth Analysis of plq functions.

3.2 Implementation: Convex univariate plq Functions

Our goal in this research is to develop a software that computes and visualizes $\partial_\varepsilon f(\bar{x})$ at an arbitrary point $\bar{x}$ and $\varepsilon > 0$ for a proper convex plq function. As visualization is a key goal, we shall focus on univariate functions.

Remark 3.1 Suppose $f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is a proper function. Then, f is a plq function if and only if it can be represented in the form

$$f(x) = \begin{cases} q_0(x) = a_0x^2 + b_0x + c_0, & \text{if } -\infty < x < x_0 \\ q_1(x) = a_1x^2 + b_1x + c_1, & \text{if } x_0 \le x \le x_1 \\ q_2(x) = a_2x^2 + b_2x + c_2, & \text{if } x_1 \le x \le x_2 \\ \vdots & \vdots \\ q_{N-1}(x) = a_{N-1}x^2 + b_{N-1}x + c_{N-1}, & \text{if } x_{N-2} \le x \le x_{N-1} \\ q_N(x) = a_Nx^2 + b_Nx + c_N, & \text{if } x_{N-1} < x < +\infty, \end{cases} \tag{5}$$

where, $a_i \in \mathbb{R}$ for $i = \{0, 1, \cdots, N\}$, $b_i \in \mathbb{R}$ for $i = \{0, 1, \cdots, N\}$, $c_i \in \mathbb{R}$ for $i = \{1, \cdots, N-1\}$ and $c_i \in \mathbb{R} \cup \{+\infty\}$ for $i = \{0, N\}$.

An interesting property of plq functions is that they are closed under many basic operations in convex analysis: Fenchel conjugation, addition, scalar multiplication, and taking the Moreau envelope [22, Proposition 5.1].

Remark 3.2 Even though the minimum of two plq functions is not necessarily a plq function (see Example 3.2), we can still compute the ε-subdifferential from the plq data structure explained in Section 3.2.1.

Example 3.2 Consider $f_1(x) = 0$ if $x \in [-1,1]$ and $+\infty$ otherwise; and $f_2(x) = x$. Clearly f_1 and f_2 are proper convex plq functions but $f(x) = \min\{f_1(x), f_2(x)\} = x$ if $x < 0$, 0 if $0 \leq x \leq 1$, and x when $1 < x$. Notice f is discontinuous at $x = 1$ as shown by Figure 3.2.

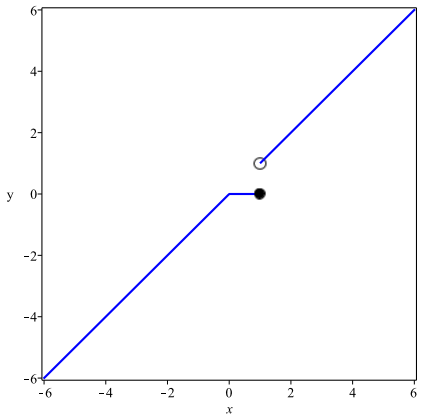


Fig. 3.2: The function $f(x) = \min(\iota_{[-1,1]}(x), x)$ is discontinuous at $x = 1$.

To implement Algorithm 1, for univariate convex plq functions, we shall use the Computational Convex Analysis (CCA) toolbox, which is openly available for download at [21]. It is coded using Scilab, a numerical software freely available [30]. The toolbox encompasses many algorithms to compute fundamental convex transforms of univariate plq functions, as introduced in [22]. Table 3.1 outlines the operations available in the CCA toolbox important to this work.

Table 3.1: Functions in the CCA Toolbox relevant to Algorithm 1

Function	**Description**
plq_check(plq$f$)	Checks integrity of a plq function
plq_isConvex(plq$f$)	Checks convexity of a plq function
plq_lft(plq$f$)	Fenchel conjugate of a plq function
plq_min(plqf_1,plqf_2)	Minimum of two plq functions
plq_isEqual(plqf_1,plqf_2)	Checks equality of two plq functions
plq_eval(plq$f$,X)	Evaluates a plq function on the grid X

3.2.1 Data Structure

We next shed some light on the data structure used in the CCA library. The CCA toolbox stores a plq function as an $(N+1)\times 4$ matrix, where each row represents one interval on which the function is quadratic.

For example, the plq function $f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ defined by (5) is stored as

$$\text{plq}f = \begin{bmatrix} x_0 & a_0 & b_0 & c_0 \\ x_1 & a_1 & b_1 & c_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{N-1} & a_{N-1} & b_{N-1} & c_{N-1} \\ +\infty & a_N & b_N & c_N \end{bmatrix}. \tag{6}$$

Note that, if $c_0 = +\infty$ or $c_N = +\infty$, then the structure demands that $a_0 = b_0 = 0$ or $a_N = b_N = 0$ respectively. If $f(x)$ is a simple quadratic function, then $N = 0$ and $x_0 = +\infty$. Finally, the special case of $f(x)$ being a shifted indicator function of a single point $\tilde{x} \in \mathbb{R}$,

$$f(x) = \iota_{\{\tilde{x}\}} + c = \begin{cases} c, & x = \tilde{x} \\ +\infty, & x \neq \tilde{x} \end{cases}$$

where $c \in \mathbb{R}$, is stored as a single row vector $\text{plq}f = \begin{bmatrix} \tilde{x} & 0 & 0 & c \end{bmatrix}$.

Remark 3.3 Throughout this paper, we shall designate f and plq$f$ for the mathematical function and the corresponding plq matrix representation.

3.2.2 The plq_epssub Algorithm

Following the plq data structure we rewrite Algorithm 1 for the specific class of univariate convex plq functions. Prior to presenting the algorithm we establish its validity.

Theorem 3.1 *Let $f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ be a univariate convex plq function, $\bar{x} \in \text{dom}(f)$ and $\varepsilon > 0$. Let $\partial_\varepsilon f(\bar{x}) = [v_l, v_u]$. Then one of the following hold.*

1. *If* $\text{plq}f^* = \begin{bmatrix} s_0 & \tilde{a}_0 & \tilde{b}_0 & \tilde{c}_0 \end{bmatrix}$ *and $s_0 \in \mathbb{R}$, then $f^*(s) = \iota_{\{s_0\}}(s) + \tilde{c}_0$ and $\tilde{a}_0 = \tilde{b}_0 = 0$; so* $\text{dom}(f^*) = \{s_0\}$ *and*

$$v_l = v_u = s_0.$$

   In this case, f must be a linear function, i.e., $f(x) = \langle s_0, x \rangle - \tilde{c}_0$ and $\tilde{c}_0 \in \mathbb{R}$.
2. *Otherwise, let*

$$\text{plq}l = \begin{bmatrix} +\infty & 0 & \bar{x} & (\varepsilon - f(\bar{x})) \end{bmatrix}$$

   and

$$\text{plq}m = \begin{bmatrix} \hat{s}_0 & \hat{a}_0 & \hat{b}_0 & \hat{c}_0 \\ \hat{s}_1 & \hat{a}_1 & \hat{b}_1 & \hat{c}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \hat{s}_{k-1} & \hat{a}_{k-1} & \hat{b}_{k-1} & \hat{c}_{k-1} \\ \hat{s}_k & \hat{a}_k & \hat{b}_k & \hat{c}_k \end{bmatrix}$$

   be the respective plq representations of $l_{\bar{x}}(s) = \varepsilon - f(\bar{x}) + \langle s, \bar{x} \rangle$ and $m(s) = \min\{f^*(s), l_{\bar{x}}(s)\}$. Then the following situations hold.

   (a) If $k = 0$, then $\hat{s}_0 = +\infty$ and

$$v_l = -\infty, \quad v_u = +\infty.$$

   In this case, f must be the indicator function of $\bar{x}$ plus a constant, i.e., $f(x) = \iota_{\{\bar{x}\}} - \hat{c}_0$ and $\hat{c}_0 \in \mathbb{R}$.

(b) If $k \geq 1$, then $\hat{s}_0 \in \mathbb{R}, \hat{s}_{k-1} \in \mathbb{R}$,

$$v_l = \begin{cases} \hat{s}_0, & \text{if} \quad \begin{bmatrix} \hat{a}_0 & \hat{b}_0 & \hat{c}_0 \end{bmatrix} = \begin{bmatrix} 0 & \bar{x} & (\varepsilon - f(\bar{x})) \end{bmatrix} \\ -\infty, & \text{otherwise} \end{cases}$$

and

$$v_u = \begin{cases} \hat{s}_{k-1}, & \text{if} \quad \begin{bmatrix} \hat{a}_k & \hat{b}_k & \hat{c}_k \end{bmatrix} = \begin{bmatrix} 0 & \bar{x} & (\varepsilon - f(\bar{x})) \end{bmatrix} \\ +\infty, & \text{otherwise} \end{cases}.$$

In order, to prove Theorem 3.1, we require the following lemmas.

Lemma 3.1 *If $f : \mathbb{R}^n \to \mathbb{R}$ has the form $f = \langle a, x \rangle + b$ where, $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, then for $\varepsilon \geq 0$ and $\tilde{x} \in \mathbb{R}^n$*

$$\partial_\varepsilon f(\tilde{x}) = \partial f(\tilde{x}) = \{\nabla f(\tilde{x})\} = \{a\}.$$

Lemma 3.2 *Let $f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ be a proper convex plq function, $\bar{x} \in \operatorname{dom}(f)$ and $\varepsilon > 0$. Define $l_{\bar{x}}(s) = \varepsilon - f(\bar{x}) + \langle s, \bar{x} \rangle$ and $m(s) = \min\{f^*(s), l_{\bar{x}}(s)\}$. Then*

(i) There exists $s \in \mathbb{R}$ such that $m(s) < l_{\bar{x}}(s)$.
(ii) We have

$$m \equiv f^* \iff f^* \equiv \langle a, \cdot \rangle + b \tag{7}$$

where $a = \bar{x}$, $b \leq \varepsilon - f(\bar{x})$, and $m \equiv f^*$ means for all s, $m(s) \equiv f^*(s)$. In this case, f must be an indicator function, $f \equiv \iota_{\{\bar{x}\}} - b$ for $b \in \mathbb{R}$, and therefore $\partial_\varepsilon f(\bar{x}) = \mathbb{R}$.

Proof We prove (i) by contradiction. Suppose $m(s) = l_{\bar{x}}(s)$, for all $s \in \mathbb{R}$, i.e. $l_{\bar{x}}(s) \leq f^*(s)$, for all $s \in \mathbb{R}$. Then using [29, Theorem 11.1] we obtain

$$f(\bar{x}) = \sup_s \{s\bar{x} - f^*(s)\} \leq \sup_s \{s\bar{x} - l(s)\} = \sup_s \{s\bar{x} - \varepsilon + f(\bar{x}) - s\bar{x}\} = f(\bar{x}) - \varepsilon$$

and that contradiction proves the lemma.

For (ii), we have

$$\begin{aligned} m \equiv f^* &\Leftrightarrow f^* \leq l_{\bar{x}}, \\ &\Leftrightarrow sx - f(x) \leq \varepsilon - f(\bar{x}) + s\bar{x}, \text{ for all } s, x, \\ &\Leftrightarrow s \in \partial_\varepsilon f(\bar{x}), \text{ for all } s, \\ &\Leftrightarrow \partial_\varepsilon f(\bar{x}) = \mathbb{R}. \end{aligned}$$

Now assume there is $y \neq x \in \operatorname{dom} f$. Then $\varepsilon + f(y) - f(\bar{x}) \geq s(y - \bar{x})$ for all s, which is not possible since the left hand-side is bounded and the right one unbounded. Hence, $\operatorname{dom} f$ is a singleton i.e. f is an indicator function. Conversely, if f is an indicator, the equivalence holds. Since the conjugate of the indicator function of a singleton is linear, we further obtain

$$m \equiv f^* \Leftrightarrow f \equiv \iota_a - b, \Leftrightarrow f^* \equiv \langle a, \cdot \rangle + b.$$

The fact that $a = \bar{x}$ is deduced from $g \equiv f^* - l_{\bar{x}} \leq 0$ (g is a convex function defined everywhere and upper bounded, hence a constant [27, Corollary 8.6.2]). The fact $b \leq \varepsilon - f(\bar{x})$ follows similarly. □

Remark 3.4 Let $f:\mathbb{R}\to\mathbb{R}\cup\{+\infty\}$ be a proper convex plq function, $\bar{x}\in\text{dom}(f)$ and $\varepsilon>0$. Let plq$m$ be the representation of $m(s)=\min\{f^*(s),l_{\bar{x}}(s)\}$. Suppose

$$\text{plq}m=\begin{bmatrix} s_0 & a_0 & b_0 & c_0\\ s_1 & a_1 & b_1 & c_1\\ \vdots & \vdots & \vdots & \vdots\\ s_{k-1} & a_{k-1} & b_{k-1} & c_{k-1}\\ s_k & a_k & b_k & c_k \end{bmatrix} \quad \text{with} \quad k\geq 1,$$

where $s_0<s_1,\ldots,s_{k-1}<s_k=+\infty$. Then, for any $i\in\{0,1,\ldots,k-1\}$

$$\begin{bmatrix} a_i & b_i & c_i\end{bmatrix}\neq\begin{bmatrix} a_{i+1} & b_{i+1} & c_{i+1}\end{bmatrix}.$$

This is because the plq_min function creates the smallest matrix representation of the minimum of two plq functions. If $[a_i\ b_i\ c_i]$ were equal to $[a_{i+1}\ b_{i+1}\ c_{i+1}]$, then the $i+1$-th row would be redundant, so not constructed by the plq_min function.

Now, we turn to the formal proof of Theorem 3.1.

Proof (of Theorem 3.1) Note that $\bar{x}\in\text{dom}(f)$ and $\varepsilon>0$ so $\partial_\varepsilon f(\bar{x})\neq\emptyset$.
Case 1 Suppose $\text{plq}f^*=\begin{bmatrix} s_0 & \tilde{a}_0 & \tilde{b}_0 & \tilde{c}_0\end{bmatrix}$ with $s_0\in\mathbb{R}$. By definition, f^* is an indicator function, i.e., $f^*(s)=\iota_{\{s_0\}}+\tilde{c}_0$. It also follows that $\tilde{a}_0=\tilde{b}_0=0$ and $\text{dom}(f^*)=\{s_0\}$. This immediately yields $f^{**}(x)=\langle s_0,x\rangle-\tilde{c}_0$. Since f is a proper convex plq function, we have that $f=f^{**}$ [29, Theorem 11.1], so $f(x)=\langle s_0,x\rangle-\tilde{c}_0$ and $\partial_\varepsilon f(\bar{x})=\{s_0\}$. Hence, $v_l=v_u=\{s_0\}$.
Case $2(a)$ Suppose

$$\text{plq}m=\begin{bmatrix} \hat{s}_0 & \hat{a}_0 & \hat{b}_0 & \hat{c}_0\\ \hat{s}_1 & \hat{a}_1 & \hat{b}_1 & \hat{c}_1\\ \vdots & \vdots & \vdots & \vdots\\ \hat{s}_{k-1} & \hat{a}_{k-1} & \hat{b}_{k-1} & \hat{c}_{k-1}\\ \hat{s}_k & \hat{a}_k & \hat{b}_k & \hat{c}_k \end{bmatrix}$$

with $k=0$. In this case,

$$\text{plq}m=\begin{bmatrix} \hat{s}_0 & \hat{a}_0 & \hat{b}_0 & \hat{c}_0\end{bmatrix}$$

then $\hat{s}_0=+\infty$. Indeed, if $\hat{s}_0\in\mathbb{R}$, then

$$\text{plq}m=\begin{bmatrix} \hat{s}_0 & 0 & 0 & \hat{c}_0\end{bmatrix}(=\iota_{\{\hat{s}_0\}}+\hat{c}_0),$$

which cannot happen as $m(s)=\min\{f^*(s),l_{\bar{x}}(s)\}\leq l_{\bar{x}}(s)<+\infty$.

To see $v_l=-\infty,v_u=+\infty$, note that from Lemma 3.2 there exists $\bar{s}\in\mathbb{R}$ such that $m(\bar{s})<l_{\bar{x}}(\bar{s})$. This implies

$$m(s)=f^*(s)\quad\text{for all}\quad s,$$

as otherwise

$$m(s)=\begin{cases} f^*(s), & s\in X_{f^*}\\ l_{\bar{x}}(s), & s\in X_l\end{cases}$$

for some $X_{f^*},X_l\subseteq\mathbb{R}$ with $X_l\neq\emptyset$. But, then $m(s)$ would be a piecewise function defined on at least two intervals contradicting $k=0$. So $m(s)=f^*(s)$ for all s, which by Lemma 3.2 gives

$$\partial_\varepsilon f(\bar{x})=\mathbb{R}.$$

Therefore, $v_l=-\infty$ and $v_u+\infty$. Consequently, from Lemma 3.2, f must be the indicator function at $\bar{x}$ plus a constant, i.e., $f(x)=\iota_{\{\bar{x}\}}-\hat{c}_0$.

Case $2(b)$ Suppose

$$\text{plq}m = \begin{bmatrix} \hat{s}_0 & \hat{a}_0 & \hat{b}_0 & \hat{c}_0 \\ \hat{s}_1 & \hat{a}_1 & \hat{b}_1 & \hat{c}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \hat{s}_{k-1} & \hat{a}_{k-1} & \hat{b}_{k-1} & \hat{c}_{k-1} \\ \hat{s}_k & \hat{a}_k & \hat{b}_k & \hat{c}_k \end{bmatrix}$$

with $k \geq 1$. By definition, $\hat{s}_0, \hat{s}_{k-1} \in \mathbb{R}$. We consider two subcases to prove the formula for v_l.
Subcase (i) Suppose $\begin{bmatrix} \hat{a}_0 & \hat{b}_0 & \hat{c}_0 \end{bmatrix} = \begin{bmatrix} 0 & \bar{x} & (\varepsilon - f(\bar{x})) \end{bmatrix}$.
By definition, $m(s) = l_{\bar{x}}(s)$ for all $s < \hat{s}_0$. From Remark 3.4 we have

$$m(s) = f^*(s) \quad \text{for all} \quad \hat{s}_0 \leq s \leq \hat{s}_1.$$

Therefore,

$$\inf\{v \in \text{dom}(f^*) : m(v) = f^*(v)\} = \hat{s}_0.$$

So $v_l = \hat{s}_0$.
Subcase (ii) Suppose $\begin{bmatrix} \hat{a}_0 & \hat{b}_0 & \hat{c}_0 \end{bmatrix} \neq \begin{bmatrix} 0 & \bar{x} & (\varepsilon - f(\bar{x})) \end{bmatrix}$.
By definition, we have

$$m(s) = f^*(s) \quad \text{for all} \quad s < \hat{s}_0.$$

Therefore,

$$\inf\{v \in \text{dom}(f^*) : m(v) = f^*(v)\} = -\infty.$$

So $v_l = -\infty$.
The formula for v_u can be proven analogously to that of v_l.

□

The details of the computation of the ε-subdifferential of a univariate convex plq function are presented in Algorithm 2. Before looking into the complexity of the algorithm, we note a minor difference between the algorithm implementation and Theorem 3.1.

Remark 3.5 In our implementation, the Case $2(a)$ of Theorem 3.1 is coded by detecting if f is an indicator function. That is, if $\text{plq}f = \begin{bmatrix} x_0 & 0 & 0 & c_0 \end{bmatrix}$, where $x_0 \in \mathbb{R}$, then $v_l = -\infty$ and $v_u = +\infty$ (Lemma 3.2) without computing plq_min.

3.2.3 Complexity of Algorithm 2

In order to prove the complexity of Algorithm 2, we require the following lemma.

Lemma 3.3 *If plq$f$ has (N+1) rows then plqf^* has $O(N)$ rows.*

Proof Since plq_lft algorithm is developed to independently operate on $(N+1)$ rows [22, Table 2] and has complexity of $O(N)$ [11, Table 2], therefore the size of the output plqf^* cannot exceed $O(N)$.

□

We now turn to the complexity of Algorithm 2.

Proposition 3.2 *If plq$f$ has (N+1) rows then Algorithm 2 runs in $O(N)$ time and space.*

Proof Table 3.2 summarizes the complexity of the independent subroutines in Algorithm 2 as stated in [22, Table 2], [11, Table 2] and the function description in Scilab.
Thus, from Table 3.2 and Lemma 3.3 the result follows.

Algorithm 2 plq_epssub Algorithm

Input: $\text{plq}f = \begin{bmatrix} x_0 & a_0 & b_0 & c_0 \\ x_1 & a_1 & b_1 & c_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{N-1} & a_{N-1} & b_{N-1} & c_{N-1} \\ +\infty & a_N & b_N & c_N \end{bmatrix}$, $\bar{x}$, $\varepsilon > 0$

Output: $\hat{v}_l, \hat{v}_u$

1: Compute plq_check(plq$f$)
if false **return** 'the input function is not plq.'

2: Compute plq_isConvex(plq$f$)
if false **return** 'the input function is not convex.'

3: Compute plq_eval(plq$f$,$\bar{x}$)
if $+\infty$, **return** '$\bar{x}$ is not in the domain of the function.';

4: **if** plqf= $\begin{bmatrix} x_0 & 0 & 0 & c_0 \end{bmatrix}$ then **return** $v_l = -\infty$, $v_u = +\infty$;

5: Compute plqf^* = plq_lft(plq$f$):

$$\text{plq}f^* = \begin{bmatrix} s_0 & \tilde{a}_0 & \tilde{b}_0 & \tilde{c}_0 \\ s_1 & \tilde{a}_1 & \tilde{b}_1 & \tilde{c}_1 \\ \vdots & \vdots & \vdots & \vdots \\ s_{\tilde{N}-1} & \tilde{a}_{\tilde{N}-1} & \tilde{b}_{\tilde{N}-1} & \tilde{c}_{\tilde{N}-1} \\ +\infty & \tilde{a}_{\tilde{N}} & \tilde{b}_{\tilde{N}} & \tilde{c}_{\tilde{N}} \end{bmatrix}.$$

if plq$f^* = \begin{bmatrix} s_0 & \tilde{a}_0 & \tilde{b}_0 & \tilde{c}_0 \end{bmatrix}$ and $s_0 \in \mathbb{R}$ **return** $\hat{v}_l = s_0, \hat{v}_u = s_0$;

6: Define

$$\text{plq}l = \begin{bmatrix} +\infty & 0 & \bar{x} & (\varepsilon - \text{plq\_eval}(\text{plq}f, \bar{x})) \end{bmatrix}.$$

7: Compute plq$m$ = plq_min(plqf^*, plq$l$):

$$\text{plq}m = \begin{bmatrix} \hat{s}_0 & \hat{a}_0 & \hat{b}_0 & \hat{c}_0 \\ \hat{s}_1 & \hat{a}_1 & \hat{b}_1 & \hat{c}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \hat{s}_{k-1} & \hat{a}_{k-1} & \hat{b}_{k-1} & \hat{c}_{k-1} \\ +\infty & \hat{a}_k & \hat{b}_k & \hat{c}_k \end{bmatrix},$$

where $\hat{s}_0 < \hat{s}_1, \ldots, < \hat{s}_{k-1} < \hat{s}_k = +\infty$.

8: Compute $\hat{v}_l, \hat{v}_u$:
If $k = 0$,

$$\hat{v}_l = -\infty, \quad \hat{v}_u = +\infty$$

If $k \geq 1$,

$$\hat{v}_l = \begin{cases} \hat{s}_0, & \text{if} \quad \begin{bmatrix} \hat{a}_0 & \hat{b}_0 & \hat{c}_0 \end{bmatrix} = \begin{bmatrix} 0 & \bar{x} & (\varepsilon - \text{plq\_eval}(\text{plq}f, \bar{x})) \end{bmatrix} \\ -\infty, \text{ otherwise} \end{cases}$$

and

$$\hat{v}_u = \begin{cases} \hat{s}_{k-1}, \text{ if} & \begin{bmatrix} \hat{a}_k & \hat{b}_k & \hat{c}_k \end{bmatrix} = \begin{bmatrix} 0 & \bar{x} & (\varepsilon - \text{plq\_eval}(\text{plq}f, \bar{x})) \end{bmatrix} \\ +\infty, & \text{otherwise} \end{cases}$$

□

Table 3.2: Core subroutines in Algorithm 2 and their complexity

Function	**Complexity**	**Variable Description**
plq_check(plq$f$)	$O(N)$	$N=$ number of rows in plq$f$
plq_isConvex(plq$f$)	$O(N)$	
plq_lft(plq$f$)	$O(N)$	
plq_min(plqf_1,plqf_2)	$O(N_1+N_2)$	$N_1,N_2=$ number of rows in plqf_1, plqf_2
plq_eval(plq$f$,X)	$O(N+\tilde{k})$	$\tilde{k}=$ number of points plq$f$ is evaluated at

4 Numerical Examples

We now present several examples which demonstrate how Algorithm 2 can be used to visualize the ε-subdifferential of univariate convex plq functions. The algorithm has been implemented in Scilab [30].

4.1 Computing $\partial_\varepsilon f(\bar{x})$ for fixed $\varepsilon > 0$ and varying $\bar{x}$

Example 4.1 Let

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

at $\bar{x} = 0$ and $\varepsilon = 1$. In plq format f is stored as

$$\text{plq}f = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ +\infty & 0 & 0 & 0 \end{bmatrix}.$$

Using plq_lft we obtain

$$\text{plq}f^*(s) = \begin{bmatrix} 0 & 0.5 & 0 & 0 \\ +\infty & 0 & 0 & +\infty \end{bmatrix} \text{ corresponding to } f^*(s) = \begin{cases} s^2/2, & s < 0 \\ +\infty, & 0 \le s \end{cases}.$$

Here $l_{\bar{x}}(s) = 1 - f(0) + \langle 0, s\rangle = 1$, in plq format we have plq$l = \begin{bmatrix} +\infty & 0 & 0 & 1 \end{bmatrix}$. Next, we compute plq_min(plqf^*,plq$l$), we obtain

$$\text{plq\_min}(\text{plq}f^*,\text{plq}l) = \begin{bmatrix} -1.4142136 & 0 & 0 & 1 \\ 0 & 0.5 & 0 & 0 \\ +\infty & 0 & 0 & 1 \end{bmatrix} \left(= \begin{cases} f^*(s), & s \in [-1.4142136, 0] \\ l_{\bar{x}}(s), & s \notin [-1.4142136, 0] \end{cases} \right).$$

Hence, we obtain $\partial_\varepsilon f(\bar{x}) \approx [-1.414, 0]$ as visualized in Figure 4.1.

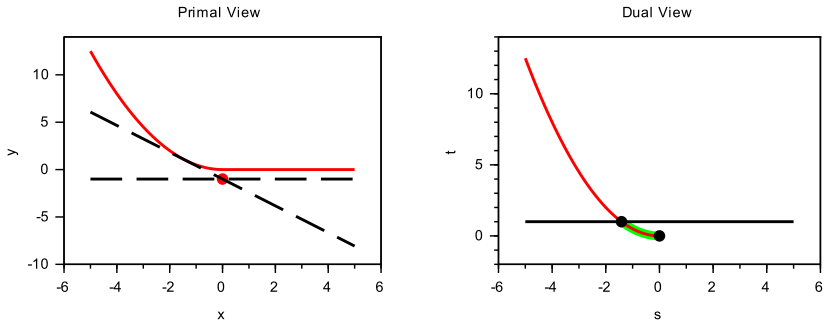


Fig. 4.1: The primal view (left) depicts the graph of $f(x) = x^2/2$ if $x < 0$, and 0 otherwise (red curve) along with the black dashed lines passing through the point $(\bar{x}, f(\bar{x}) - \varepsilon)$ (red dot), with $\bar{x} = 0$ and $\varepsilon = 1$, having slopes -1.414 and 0 respectively (the lower and upper bounds of $\partial_\varepsilon f(\bar{x})$). The dual view depicts the graphs of $f^*(s)$ (red curve) and $l_{\bar{x}}(s)$ (solid black line). The green curve shows when the graphs of $m(s)$ and $f^*(s)$ coincide.

In one dimension, from Figure 4.1, we may geometrically interpret that the ε-subdifferential set consists of all possible slopes belonging to the interval $[-1.414, 0]$, resulting in all possible lines with the respective slopes passing through the point $(\bar{x}, f(\bar{x}) - \varepsilon) = (0, -1)$.

We now look into visualizing the multifunction $x \mapsto \partial_\varepsilon f(x)$ for a given $\varepsilon > 0$.

Example 4.2 We consider

$$f(x) = \begin{cases} -7x - 5, & -\infty < x < -1 \\ x^2 - x, & -1 \leq x \leq 1 \\ 2x^2 - 3x + 1, & 1 < x < \infty \end{cases},$$

and $\varepsilon = 1$. As seen in Figure 4.2, for $\bar{x} = -1$, $\partial_\varepsilon f(\bar{x}) = [-7, -1]$. Correspondingly, for the choice of $\bar{x} = 0.5$ we obtain $\partial_\varepsilon f(\bar{x}) \approx [-2, 2.162]$, as visualized in Figure 4.3.

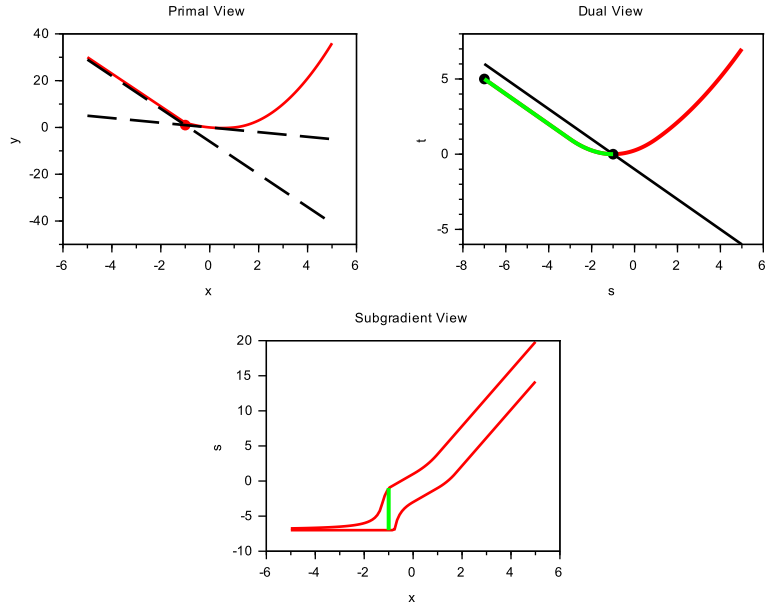


Fig. 4.2: The primal view (left) depicts the graph of $f(x) = -7x - 5$ if $x < -1$, $f(x) = x^2 - x$ if $-1 \leq x \leq 1$ and $2x^2 - 3x + 1$ otherwise (red curve) along with the black dashed lines passing through the point $(\bar{x}, f(\bar{x}) - \varepsilon)$ (red dot), with $\bar{x} = -1$ and $\varepsilon = 1$, having slopes -7 and -1 respectively (the lower and upper bounds of $\partial_\varepsilon f(\bar{x})$). The dual view (right) depicts the graphs of $f^*(s)$ (red curve) and $l_{\bar{x}}(s)$ (solid black line). The green curve shows when the graphs of $m(s)$ and $f^*(s)$ coincide. The subgradient view (bottom) shows the graph of $x \mapsto \partial_\varepsilon f(\bar{x})$ (lower and upper bounds) with the green line showing $\partial_\varepsilon f(-1)$.

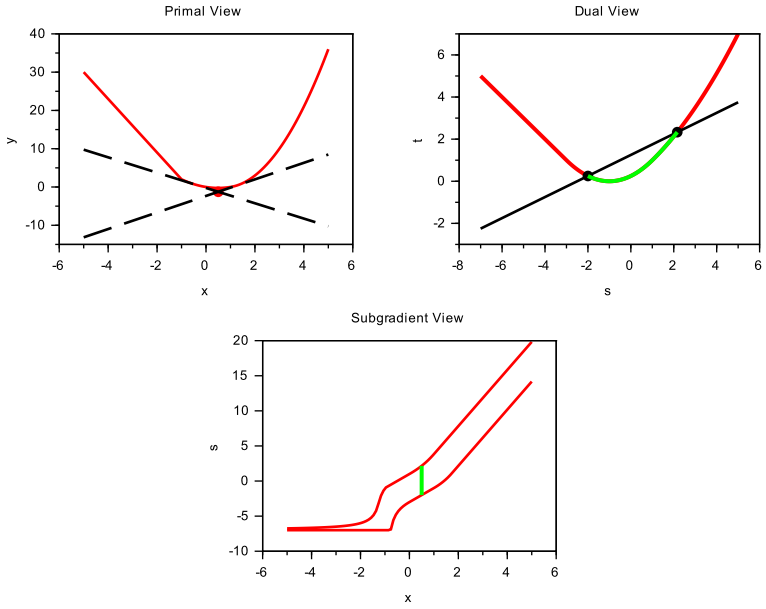


Fig. 4.3: The primal view (left) depicts the graph of $f(x) = -7x - 5$ if $x < -1$, $f(x) = x2 - x$ if $-1 \leq x \leq 1$ and $2x^2 - 3x + 1$ otherwise (red curve) along with the black dashed lines passing through the point $(\bar{x}, f(\bar{x}) - \varepsilon)$ (red dot), with $\bar{x} = 0.5$ and $\varepsilon = 1$, having slopes -2 and 2.162 respectively (the lower and upper bounds of $\partial_\varepsilon f(\bar{x})$). The dual view (right) depicts the graphs of $f^*(s)$ (red curve) and $l_{\bar{x}}(s)$ (solid black line). The green curve shows when the graphs of $m(s)$ and $f^*(s)$ coincide. The subgradient view (bottom) shows the graph of $x \mapsto \partial_\varepsilon f(\bar{x})$ (lower and upper bounds) with the green line showing $\partial_\varepsilon f(0.5)$.

In Figure 4.3, the graph of $\partial_\varepsilon f(x)$ (red curve) as a function of $x \in [-5,5]$ with $\varepsilon = 1$ is sketched by iteratively computing the respective lower and the upper bounds of $\partial_\varepsilon f(x)$ for 100 equally spaced points in the interval $[-5,5]$. This process takes under 5 seconds on a basic computer.

Example 4.3 Let

$$f(x) = \begin{cases} +\infty, & x < -6 \\ -2x, & -6 \leq x \leq 0 \\ x^2 - 2x, & 0 < x \leq 2 \\ 2x - 4, & 2 < x \leq 3 \\ \frac{1}{3}x^2 - 1, & 3 < x \end{cases}.$$

An animated visualization of $\partial_\varepsilon f(\bar{x})$ for the example is presented in the following Figure 4.4, that takes into account $\varepsilon = 1$ and the choices of $\bar{x}$ as 50 equally spaced points between $[-5, 4.5]$.

4.2 Computing $\partial_\varepsilon f(\bar{x})$ for fixed $\bar{x}$ and varying $\varepsilon > 0$

We can also visualize the graph of $\partial_\varepsilon f(\bar{x})$ as a function of $\varepsilon > 0$ for a given $\bar{x}$.

Example 4.4 Consider $\bar{x} = 0$ and let

$$f(x) = \begin{cases} \frac{1}{6}x^2 + \frac{1}{3}x, & x < -2 \\ x + 2, & -2 \leq x \leq 1 \\ +\infty, & 1 < x \end{cases}.$$

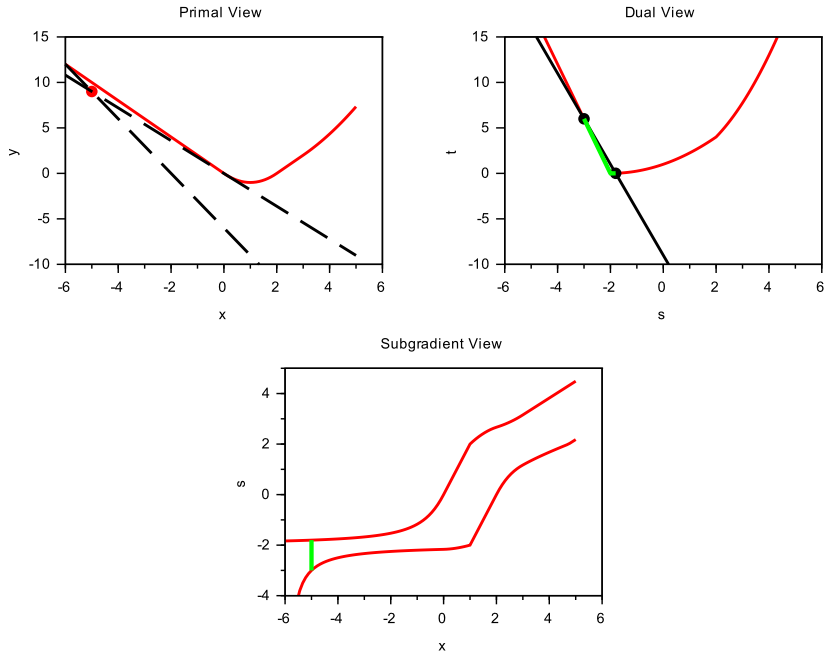


Fig. 4.4: Animated version of the primal view (left), dual view (right), and subdifferential graph (bottom) for $f(x) = +\infty$ if $x < -6$, $f(x) = -2x$ if $-6 \le x \le 0$, $f(x) = x^2 - 2x$ if $0 \le x \le 2$, $f(x) = 2x - 5$ if $2 \le x \le 3$ and $f(x) = x^2/3 - 1$ otherwise (red curve) along with the black dashed lines passing through the point $(\bar{x}, f(\bar{x}) - \varepsilon)$ (red dot) for$\varepsilon = 1$. The slopes are the lower and upper bounds of $\partial_\varepsilon f(\bar{x})$. The dual view (right) depicts the graphs of $f^*(s)$ (red curve) and $l_{\bar{x}}(s)$ (solid black line). The green curve shows when the graphs of $m(s)$ and $f^*(s)$ coincide. The subgradient view (bottom) shows the graph of $x \mapsto \partial_\varepsilon f(\bar{x})$ (lower and upper bounds) with the green line showing $\partial_\varepsilon f(\bar{x})$.

An animated visualization of $\partial_\varepsilon f(\bar{x})$ for the example is presented in the following Figure 4.5, that takes into account $\bar{x} = -1$ and the choices of $\varepsilon > 0$ as 50 equally spaced points between $[0.1, 3]$.

4.3 An illustration of Brøndsted-Rockafellar Theorem

In this section, we visualize the Brøndsted-Rockafellar theorem.

Theorem 4.1 *[14, Theorem XI.4.2.1] Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lower- semicontinuous convex function, $\bar{x} \in \mathrm{dom}(f)$ and $\varepsilon \ge 0$. For any $\lambda > 0$ and $s \in \partial_\varepsilon f(\bar{x})$, there exists $\bar{x}_\lambda \in \mathrm{dom}(f)$ and $s_\lambda \in \partial f(\bar{x}_\lambda)$ such that $\|\bar{x}_\lambda - \bar{x}\| \le \lambda$ and $\|s_\lambda - s\| \le \varepsilon/\lambda$.*

Theorem 4.1 asserts that for a one-dimensional proper lower-semicontinuous convex function, any ε-subgradient at $\bar{x}$ can be approximated by some true subgradient computed (possibly) at some $y \ne \bar{x}$, lying within a rectangle of width λ and height ε/λ. For a better understanding, we consider the following animated example.

Example 4.5 Consider for $x \in \mathbb{R}$

$$f(x) = \begin{cases} x^2/3, & x \le -2, \\ x/2 + 7/3, & -2 \le x \le 2.5, \\ x^2 - 8/3, & 2.5 < x, \end{cases}$$

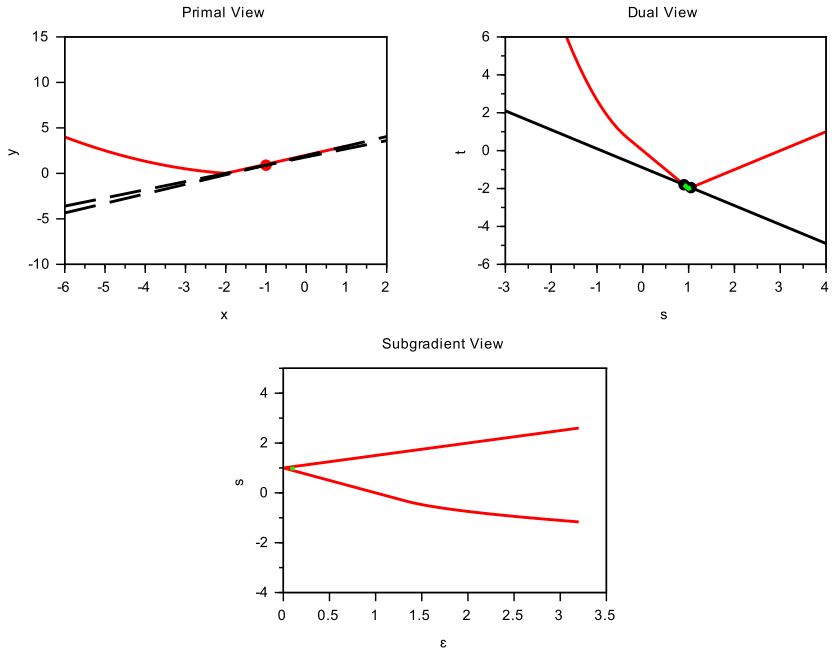
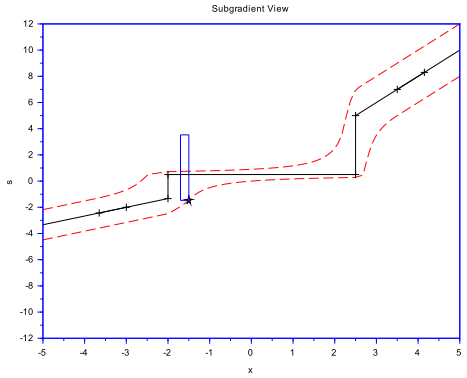
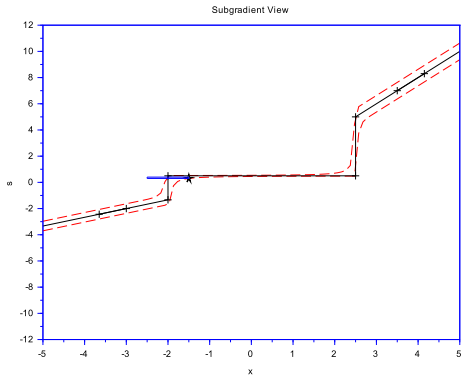


Fig. 4.5: Animated version of the primal view (left), dual view (right), and subdifferential graph (bottom) for $f(x) = x^2/6 + x/3$ if $x < -2$, $f(x) = x + 2$ if $-2 \leq x \leq 1$, $f(x) = +\infty$ otherwise (red curve) along with the black dashed lines passing through the point $(\bar{x}, f(\bar{x}) - \varepsilon)$ (red dot) for$\bar{x} = -1$. The slopes are the lower and upper bounds of $\partial_\varepsilon f(\bar{x})$. The dual view (right) depicts the graphs of $f^*(s)$ (red curve) and $l_{\bar{x}}(s)$ (solid black line). The green curve shows when the graphs of $m(s)$ and $f^*(s)$ coincide. The subgradient view (bottom) shows the graph of $\varepsilon \mapsto \partial_\varepsilon f(\bar{x})$ (lower and upper bounds) with the green line showing $\partial_\varepsilon f(-1)$.

and $\bar{x} = -1.5$. Figure 4.1 visualizes the Brøndsted-Rockafellar theorem at $(\bar{x}, v_l) \approx (-1.5, -1.471) \in \partial_\varepsilon f(-1.5)$ (black star).



(a) For $\varepsilon = 1$ and $\lambda \in [0.2, 2]$



(b) For $\lambda = 1$ and $\varepsilon \in [0.1, 2]$

Fig. 4.6: An illustration of Brøndsted-Rockafellar theorem at $\bar{x} = -1.5$. The graph depicts $\partial_\varepsilon f(x)$ (red dashed curve) and $\partial f(x)$ (black curve) for $x \in [-5, 5]$. The theorem asserts that the blue rectangle always intersect the black curve.

In Figure 4.6a, for a given $\varepsilon = 1$, and $(\bar{x}, v_l) \approx (-1.5, -1.471)$, we plot rectangles having respective dimensions $\lambda \times (\varepsilon/\lambda)$ for 50 different choices of $\lambda \in [0.2, 2]$. We observe that, as stated by the Brøndsted-Rockafellar theorem, for each choice of λ the rectangles intersect the true subdifferential. Likewise, in Figure 4.6*b* we repeat the same process with a fixed $\lambda = 1$ and 50 different choices of $\varepsilon \in [0.1, 2]$ leading to a similar conclusion.

5 Conclusion and Future Work

In this work, we first proposed a general algorithm that computes the ε-subdifferential of any proper convex function, and then presented an implementation for univariate convex plq functions. The implementation allows for rapid computation and visualization of the ε-subdifferential for any such function, and extends the CCA numerical toolbox.

Noting that the algorithm is implementable in one dimension, it is natural to ask whether an extension to higher dimension is possible. Note that if $f : \mathbb{R}^n \to \mathbb{R}$, then visualization of the subdifferential is

difficult, since $\partial f(x)$ is a set-valued mapping from $\mathbb{R}^n$ into $\mathbb{R}^n$. In addition, in dimensions greater than 1, the minimum of two plq functions is no longer representable using the plq data structure presented in Subsection 3.2.1. Consider, for example,

$$\min(1/2\|s\|^2 - 1, 0) = \begin{cases} 1/2\|s\|^2 - 1 & \|x\| \leq 1, \\ 0 & \|x\| > 1. \end{cases}$$

and note that the domain is not split into polyhedral pieces. Hence, the ε-subdifferential is no longer polyhedral even for functions of 2 variables. It is a convex set whose boundary is defined by piecewise curves; in some cases it is an ellipse.

Note that there has been work on computing the conjugate of bivariate functions [17,11,9]. However, the resulting data structures are much harder to manipulate. We leave it to future work to extend our results to higher dimensions.

Two other directions for future work are as follows. First, the current method to produce the subdifferential view (e.g., Figure 4.2) requires computing the ε-subdifferential for a wide selection of x values. It may be possible to improve this through a careful analysis of Proposition 3.1. Another clearly valuable direction of extension would be developing methods to visualize the ε-subdifferential of any univariate convex function. A first approach to this could be achieved by approximating the univariate convex function with a univariate convex plq function. However, it may be more efficient to try to directly solve $\inf\{v \in \mathrm{dom}(f^*) : m(v) = f^*(v)\}$ and $\sup\{v \in \mathrm{dom}(f^*) : m(v) = f^*(v)\}$ using a numerical optimization method.

Acknowledgements This work was supported in part by Discovery Grants #355571-2013 (Hare) and #298145-2013 (Lucet) from NSERC, and The University of British Columbia, Okanagan campus. Part of the research was performed in the Computer-Aided Convex Analysis (CA2) laboratory funded by a Leaders Opportunity Fund (LOF) from the Canada Foundation for Innovation (CFI) and by a British Columbia Knowledge Development Fund (BCKDF).

Special thanks to Heinz Bauschke for recommending we visualize the Brøndsted-Rockafellar Theorem.

References

1. Aravkin, A., Burke, J., Pillonetto, G.: Sparse/robust estimation and Kalman smoothing with nonsmooth log-concave densities: Modeling, computation, and theory. Journal of Machine Learning Research **14**, 2689–2728 (2013). URL http://jmlr.org/papers/v14/aravkin13a.html
2. Bonnans, J., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A.: Numerical Optimization: Theoretical and Practical Aspects. Springer Science & Business Media (2006)
3. Borwein, J., Lewis, A.: Convex Analysis and Nonlinear Optimization: Theory and Examples. Springer Science & Business Media (2010)
4. Brøndsted, A., Rockafellar, R.T.: On the subdifferentiability of convex functions. Proc. Amer. Math. Soc. **16**, 605–611 (1965). URL http://www.jstor.org/stable/2033889
5. Correa, R., Hantoute, A., Jourani, A.: Characterizations of convex approximate subdifferential calculus in Banach spaces. Trans. Amer. Math. Soc. **368**(7), 4831–4854 (2016). DOI 10.1090/tran/6589. URL http://dx.doi.org/10.1090/tran/6589
6. Correa, R., Lemaréchal, C.: Convergence of some algorithms for convex minimization. Math. Programming **62**(2, Ser. B), 261–275 (1993). DOI 10.1007/BF01585170. URL http://dx.doi.org/10.1007/BF01585170
7. Dembo, R., Anderson, R.: An efficient linesearch for convex piecewise-linear/quadratic functions. In: Advances in numerical partial differential equations and optimization (Mérida, 1989), pp. 1–8. SIAM, Philadelphia, PA (1991)
8. Frangioni, A.: Generalized bundle methods. SIAM J. Optim. **13**(1), 117–156 (electronic) (2002). DOI 10.1137/S1052623498342186. URL http://dx.doi.org/10.1137/S1052623498342186
9. Gardiner, B., Jakee, K., Lucet, Y.: Computing the partial conjugate of convex piecewise linear-quadratic bivariate functions. Comput. Optim. Appl. **58**(1), 249–272 (2014). DOI 10.1007/s10589-013-9622-z. URL http://dx.doi.org/10.1007/s10589-013-9622-z
10. Gardiner, B., Lucet, Y.: Convex hull algorithms for piecewise linear-quadratic functions in computational convex analysis. Set-Valued and Variational Analysis **18**(3-4), 467–482 (2010)
11. Gardiner, B., Lucet, Y.: Computing the conjugate of convex piecewise linear-quadratic bivariate functions. Mathematical Programming **139**(1-2), 161–184 (2013). DOI 10.1007/s10107-013-0666-8. URL http://dx.doi.org/10.1007/s10107-013-0666-8

12. Hare, W., Planiden, C.: Thresholds of prox-boundedness of PLQ functions. Journal of Convex Analysis **23**(3), 1–28 (2016)
13. Hare, W., Sagastizábal, C.: A redistributed proximal bundle method for nonconvex optimization. SIAM J. Optim. **20**(5), 2442–2473 (2010). DOI 10.1137/090754595. URL http://dx.doi.org/10.1137/090754595
14. Hiriart-Urruty, J., Lemaréchal, C.: Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods, vol. 306 of Grundlehren der mathematischen Wissenschaften. Springer-Verlag, New York (1993)
15. Hiriart-Urruty, J., Moussaoui, M., Seeger, A., Volle, M.: Subdifferential calculus without qualification conditions, using approximate subdifferentials: A survey. Nonlinear Analysis: Theory, Methods & Applications **24**(12), 1727–1754 (1995)
16. Ioffe, A.D.: Approximate subdifferentials and applications. I. The finite-dimensional theory. Trans. Amer. Math. Soc. **281**(1), 389–416 (1984). URL http://dx.doi.org/10.2307/1999541
17. Jakee, K.M.K.: Computational convex analysis using parametric quadratic programming. Master's thesis, University of British Columbia (2013). URL https://circle.ubc.ca/handle/2429/45182
18. Kiwiel, K.: Proximity control in bundle methods for convex nondifferentiable minimization. Math. Programming **46**(1, (Ser. A)), 105–122 (1990). DOI 10.1007/BF01585731. URL http://dx.doi.org/10.1007/BF01585731
19. Kiwiel, K.: Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. Math. Programming **69**(1, Ser. B), 89–109 (1995). DOI 10.1007/BF01585554. URL http://dx.doi.org/10.1007/BF01585554. Nondifferentiable and large-scale optimization (Geneva, 1992)
20. Lemaréchal, C., Sagastizábal, C.: Variable metric bundle methods: from conceptual to implementable forms. Math. Programming **76**(3, Ser. B), 393–410 (1997). DOI 10.1016/S0025-5610(96)00053-6. URL http://dx.doi.org/10.1016/S0025-5610(96)00053-6
21. Lucet, Y.: Computational convex analysis library, 1996–2016. http://atoms.scilab.org/toolboxes/CCA/
22. Lucet, Y., Bauschke, H., Trienis, M.: The piecewise linear-quadratic model for computational convex analysis. Computational Optimization and Applications **43**(1), 95–118 (2009)
23. de Oliveira, W., Sagastizábal, C.: Level bundle methods for oracles with on-demand accuracy. Optim. Methods Softw. **29**(6), 1180–1209 (2014). DOI 10.1080/10556788.2013.871282. URL http://dx.doi.org/10.1080/10556788.2013.871282
24. de Oliveira, W., Solodov, M.: A doubly stabilized bundle method for nonsmooth convex optimization. Math. Program. **156**(1-2, Ser. A), 125–159 (2016). DOI 10.1007/s10107-015-0873-6. URL http://dx.doi.org/10.1007/s10107-015-0873-6
25. Rantzer, A., Johansson, M.: Piecewise linear quadratic optimal control. Automatic Control, IEEE Transactions on **45**(4), 629–637 (2000)
26. Rockafellar, R.: On the essential boundedness of solutions to problems in piecewise linear quadratic optimal control. Analyse mathematique et applications. Gauthier villars pp. 437–443 (1988)
27. Rockafellar, R.: Convex Analysis. Princeton University Press (2015)
28. Rockafellar, R., Wets, R.: A lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming. In: Stochastic Programming 84 Part II, pp. 63–93. Springer (1986)
29. Rockafellar, R., Wets, R.: Variational Analysis, vol. 317. Springer Science & Business Media (2009)
30. Scilab:: Scilab. http://www.scilab.org/ (2015)
31. Trienis, M.: Computational convex analysis: From continuous deformation to finite convex integration. Master thesis (2007)