ECE 602 – Section 4
Non-smooth optimization of convex functions

- Subgradient, subdifferential and their properties
- First-order optimality condition for sub-differentiable functions
- Proximal mapping and Proximal Point Algorithm
- Conjugate functions and Moreau decomposition
- Proximal Gradient Method and Douglas-Rachford Splitting

- In the previous sections, we learned a number of methods of *unconstrained smooth convex optimization* which could be used to solve

$$\min_{x} \; f(x)$$

for some convex $f : \mathbf{dom}\, f \to \mathbf{R}$ of either $\mathcal{C}^1$ or $\mathcal{C}^2$ class.

- In particular, we have discussed several *first-order methods*, *viz.*
  - Gradient Descent Method (GDM)
  - Conjugate Gradients Method (CGM)
  - Gauss-Newton Method (GNM) (for non-linear LS problems), and
  - Levenberg-Marquardt algorithm (for non-linear and possibly non-convex problems)

- In general, first-order methods share the following *pros* and *cons*:

Pros numerically "cheap" iterations (no need for $\nabla^2 f(x)$), guaranteed convergence to a (local) minimum

Cons relatively slow convergence rates

- As the next step, we want to extend our discussion to *constrained optimization problems*.
- For a given $f : \textbf{dom}\, f \to \mathbf{R}$, a constrained optimization problem can be defined as

$$\min_{x} \; f(x)$$
$$\text{subject to } x \in \mathcal{C}$$

  where we use "subject to" (often abbreviated as "s.t.") to require that the optimal solution has to be found within set $\mathcal{C}$.
- Such set is called the *set of feasible solutions*, which we always assume to be non-empty.
- When f is a convex over its domain and the feasible set $\mathcal{C}$ is closed and convex as well, the above optimization problem is referred to as *convex*.

- The constrained optimization problem $\min_{x \in \mathcal{C}} f(x)$ can be cast into an equivalent *unconstrained* form using the notion of an *indicator function*.
- Recall that, given a set $C \subset \textbf{dom}\, f$, its indicator function is defined as

$$I_{\mathcal{C}}(x) = \begin{cases} 0, & \text{if } x \in \mathcal{C} \\ \infty, & \text{otherwise} \end{cases}$$

- Assuming $\inf_x f(x) < \infty$, an *equivalent unconstrained problem* has the form of

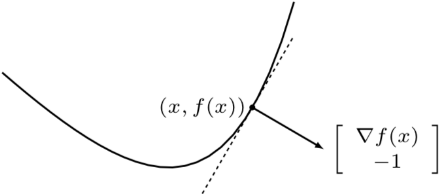
$$\min_x \; \{f(x) + I_{\mathcal{C}}(x)\}$$

- Note that, in this case, we effectively minimize $\tilde{f}(x) = f(x) + I_{\mathcal{C}}$ which takes values over the *extended real line* $(-\infty, +\infty]$.
- Such functions are called *extended-value functions*.

- To take advantage of the unconstrained formulation, we need to learn how to deal with extended-value functions.
- Fortunately, working with such functions is quite straightforward under a few standard assumptions and some additional precautions (for more details see Section 3.1.2 of Boyd's textbook).
- More importantly, the sublevel sets of $I_\mathcal{C}$, i.e. $\mathcal{S}_\alpha = \{x \mid I_\mathcal{C}(x) \leq \alpha\}$, are *convex and closed* as long as $\mathcal{C}$ is *convex and closed*.
- Hence, if f is closed and convex, so will be $\tilde{f}$. And this is what turns out to be of key importance for the algorithms of this section.
- Note however that $\tilde{f}$ is *not* differentiable, meaning that *our gradient-based tools are no longer applicable.*
- To overcome this setback, we need to exploit some tools of *non-smooth optimization* which are discussed next.

- The defining inequality for *differentiable convex functions* states

$$f(y) \geq f(x) + \nabla f(x)^T (y - x), \quad \forall y \in \textbf{dom}\, f$$



- Here the graph of f is viewed as a *parametric curve*, i.e. a map from $\textbf{dom}\, f$ to $\textbf{dom}\, f \times \mathbf{R}$, namely $x \mapsto (x, f(x))$.

- For each x, the *tangent vector* to the curve at $(x, f(x))$ is obtained by differentiating the latter w.r.t. x, resulting in $[1, \nabla f(x)]^T$.

- Consequently, the *normal vector* is defined as $[\nabla f(x), -1]^T$ (as shown in the above figure).

- The 1st-order approximation of f at x is a global lower bound (*under-estimator*), since $\nabla f(x)$ defines a non-vertical *supporting hyperplane* to $\mathbf{epi}\, f$ at point $(x, f(x))$.

- Formally, we have

$$\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix} \left(\begin{bmatrix} y \\ t \end{bmatrix} - \begin{bmatrix} x \\ f(x) \end{bmatrix} \right) \le 0, \quad \forall (y,t) \in \mathbf{epi}\, f$$

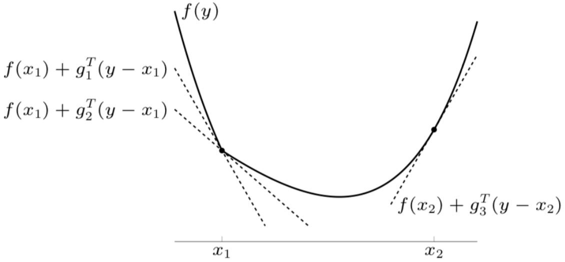
- Note that, initially, we perceived f as an *algebraic* entity, i.e. a formal mathematical rule which establishes a relation between x and $y = f(x)$.

- *Convexity* makes it possible to think of f as a *geometric* entity, namely $\mathbf{epi}\, f$, which is a convex and closed subset of $\mathbf{dom}\, f \times \mathbf{R}$, as long as f is closed and convex.

- In this case, $\mathbf{epi}\, f$ can be defined as the *intersection* of all half-spaces defined by the normals $[\nabla f(x), -1]^T$.

- Now assume that f is still *convex but not differentiable continuously everywhere in the interior of its domain*, i.e. $\textbf{int dom}\, f$.
- In this case, its *subgradient* at x is *any* vector g that satisfies

$$f(y) \geq f(x) + g^T(y - x), \quad \forall y \in \textbf{dom}\, f$$



Here, g_1, g_2 are subgradient of f at x_1, while g_3 is a subgradient of f at x_2.

- One can see, at the points of discontinuity of $f'(x)$ (or, more generally, of $\nabla f(x)$), there might be *multiple* subgradients.

- At each $x \in \mathbf{dom}\, f$, all available subgradients are combined into a set $\partial f(x)$ called the *subdifferential* of f at x. Formally,

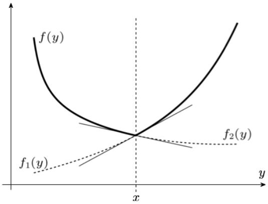
$$\partial f(x) = \left\{ g \mid g^T(y - x) \leq f(y) - f(x), \forall y \in \mathbf{dom}\, f \right\}$$

- Note that the notation $g^T(y - x)$ is more appropriate in the case when all the vectors are in $\mathbf{R}^n$. More generally, we should use $\langle g, y - x \rangle$.

- As $\partial f(x)$ is an intersection of (closed) half-spaces, it is a closed convex set. Moreover, if $x \in \mathbf{int\, dom}\, f$, then $\partial f(x)$ is nonempty and bounded.

- Consider $f(x) = \max\{f_1(x), f_2(x)\}$, with both $f_1(x)$ and $f_2(x)$ being convex and differentiable.

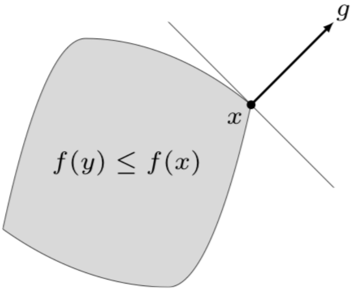
$$\partial f(y) = \begin{cases} \{\nabla f_1(y)\}, & \text{if } y > x \\ [\nabla f_1(y), \nabla f_1(y)], & \text{if } y = x \\ \{\nabla f_2(y)\}, & \text{if } y < x \end{cases}$$



- Let $x \in \mathbf{R}^n$ and consider $f(x) = \|x\|_2$. In this case,

$$\partial f(x) = \begin{cases} \{x/\|x\|_2\}, & \text{if } x \neq 0 \\ \{g \in \mathbf{R}^n \mid \|g\|_2 \leq 1\}, & \text{if } x = 0 \end{cases}$$

- Suppose x satisfies $f(x) \geq f(y)$, then this implies $g^T(y-x) \leq 0$, where g is a subgradient of f at x.



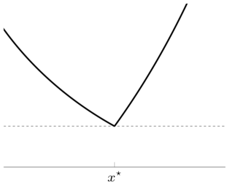
- In other words, the nonzero subgradients at x define supporting hyperplanes to the sub-level set

$$\{y \mid f(y) \leq f(x)\}$$

- Using the subdifferential, we can now define the *first-order optimality condition of non-smooth convex optimization*.

- In the *unconstrained* setting, x^* minimizes $f(x)$ *if and only if*

$$\bar{0} \in \partial f(x^*)$$



where $\bar{0}$ is a zero-vector of the same dimensionality as x.

- Note that the validity of the above statement is easy to confirm, since, by definition:

$$f(y) \geq f(x^*) + \langle \bar{0}, y - x^* \rangle, \quad \forall y,$$

and, therefore, $\bar{0} \in \partial f(x^*)$.

- Recall that, the gradient ∇f of a continuously differentiable function f can be viewed as a *linear operator* $x \mapsto \nabla f(x)$, $\forall x \in \textbf{int dom}\, f$.

- In this case, if f is also convex then, for any $x, y \in \textbf{int dom}\, f$, we have

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0$$

which characterizes the gradient operator as *monotone* (or *strictly* monotone, when f is *strictly* convex).

- Just like the gradient operator, subdifferential is also a linear operator. Moreover, it can also be shown to be monotone for convex f, *viz.*

$$\langle u - v, x - y \rangle \geq 0$$

for all $x, y \in \textbf{dom}\, f$ and $u \in \partial f(x)$, $v \in \partial f(y)$.

- Yet, in contrast to the gradient, the subdifferential map is *multivalued* or *set-valued*, since its value at x is, in fact, a set.

- Note that if f is differentiable at x, then $\partial f(x)$ is a *singleton* (i.e., a set of one element), *viz.* $\partial f(x) = \{\nabla f(x)\}$.

- To appreciate the effect of monotonicity of ∂f, let us consider

$$\min_x \left\{ \frac{1}{2}\|x - y\|_2^2 + \lambda\|x\|_1 \right\}$$

for some given $y \in \mathbf{R}^n$ and a (*regularization parameter*) $\lambda > 0$.

- First, we note that the problem is *separable* since

$$f(x) = \frac{1}{2}\|x - y\|_2^2 + \lambda\|x\|_1 = \sum_{i=1}^{n} \underbrace{\left(\frac{1}{2}|x_i - y_i|^2 + \lambda|x_i| \right)}_{\varphi_i(x_i)} = \sum_{i=1}^{n} \varphi_i(x_i)$$

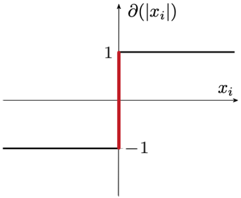
and, therefore, $f(x)$ can be minimized via *independent* minimization of $\varphi_i(x_i)$, for all $i = 1, 2, \ldots, n$.

- Thus, all we need now is to solve a *scalar* problem of the form

$$\min_{x_i} \left\{ \frac{1}{2}|x_i - y_i|^2 + \lambda|x_i| \right\}$$

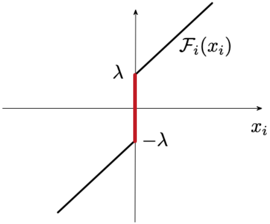
- The subdifferential of $\varphi_i(x_i)$ is given by $\partial\varphi_i(x_i) = (x_i - y_i) + \lambda\,\partial(|x_i|)$, with $\partial(|x_i|)$ defined as

$$\partial(|x_i|) = \begin{cases} 1, & \text{if } x_i > 0 \\ [-1, 1], & \text{if } x_i = 0 \\ -1, & \text{if } x_i < 0 \end{cases}$$



Note that $\partial(|x_i|)$ is a monotone function.

- Therefore, the 1st-order optimality condition suggests that



$$y_i \in \overbrace{x_i^* + \lambda\,\partial(|x_i^*|)}^{\mathcal{F}_i(x_i^*)}$$

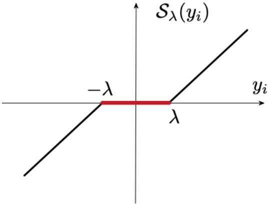
Note that function $\mathcal{F}_i$ is *strictly monotone* and *onto*, which suggests that

$$x_i^* = \mathcal{F}_i^{-1}(y_i)$$

- Due to the properties of $\mathcal{F}_i$, its inverse function $\mathcal{S}_\lambda := \mathcal{F}_i^{-1}$ is always well-defined. In particular, in the case at hand, this function is known as *soft-thresholding*.

$$x_i^* = \mathcal{S}_\lambda(y_i) = (|y_i| - \lambda)_+ \operatorname{sign}(y_i) =$$

$$= \begin{cases} y_i - \lambda, & \text{if } y_i > \lambda \\ 0, & \text{if } -\lambda \le y_i \le \lambda \\ y_i + \lambda, & \text{if } y_i < -\lambda \end{cases}$$



- The optimal solution to the original problem can be defined as

$$x^* = \mathcal{S}_\lambda(y) = [\mathcal{S}_\lambda(y_1), \mathcal{S}_\lambda(y_2), \ldots, \mathcal{S}_\lambda(y_n)]^T$$

- Note that x^* is obtained via applying $\mathcal{S}_\lambda$ to each coordinate y_i of y independently (i.e., *separably*).

- The above solution has been made possible due to the monotonicity of the subdifferential of $\|x\|_1$.

Resolvent (cont.)

- The previous examples demonstrates a number of important concepts.
- Let $f : \mathbf{R}^n \to \mathbf{R}$ be convex, closed and *sub-differentiable*. Consider an optimization problem of the form

$$\min_x \left\{ \frac{1}{2} \|x - y\|_2^2 + \lambda\, f(x) \right\}$$

  with its associated optimality condition

$$y \in \; x^* + \lambda \partial f(x^*) = (\mathcal{I} + \lambda\, \partial f)(x^*)$$

  where $\mathcal{I} + \lambda\, \partial f$ needs to be viewed as an operator from $\mathbf{R}^n$ to itself (with $\mathcal{I}$ being the identity operator).
- As $\mathcal{I} + \lambda\, \partial f$ is strictly monotone and onto (and, therefore, *injective*), its inverse $\mathcal{R}_\lambda = (\mathcal{I} + \lambda\, \partial f)^{-1}$ – called the *resolvent* of ∂f – is always well-defined.
- Moreover, the optimal solution to our optimization problem can now be defined as

$$\boxed{x^* = \mathcal{R}_\lambda(y)}$$

- When the resolvent in question pertains to the subdifferential ∂f of a convex function f, its commonly referred to as the *proximal mapping* (aka *proximal operator*) of f. Formally,

$$\mathbf{prox}_{\lambda f}(y) = \arg\min_{x} \left\{ \frac{1}{2}\|x-y\|_2^2 + \lambda f(x) \right\}$$

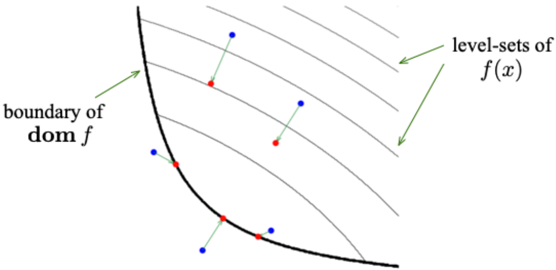
  and it is *unique* if f is closed.

- In some sense, **prox** generalizes the notion of *orthogonal projection*. Indeed, let $f(x) = I_{\mathcal{C}}(x)$, for some closed and convex set $\mathcal{C}$. Then,

$$\mathbf{prox}_{I_{\mathcal{C}}}(y) = \arg\min_{x \in \mathcal{C}} \|x-y\|_2^2 = \mathcal{P}_{\mathcal{C}}(y)$$

  which is the *orthogonal projection* of y onto $\mathcal{C}$.

- Note that, for $f(x) = 0$, $\mathbf{prox}_{\lambda f}(y) = y$.



- The above figure shows the level curves of a convex function $f(x)$ over its domain.
- Applying $\mathbf{prox}_f$ to the "blue" points moves them to the corresponding "red" points.
- The "outside" blue points move simultaneously towards the minimum of $f(x)$ *and* the boundary of **dom** f.

- Let $f : \mathbf{R}^n \to \mathbf{R}$ be a closed and convex function and let x^* be its global minimizer, i.e. $f(x^*) \leq f(x)$ for all x. Then,

$$\mathbf{prox}_f(x^*) = \arg\min_x \left\{ \frac{1}{2}\|x - x^*\|_2^2 + f(x) \right\} = x^*.$$

- Consequently, the point x^* minimizes f *if and only if*

$$\boxed{x^* = \mathbf{prox}_f(x^*)}$$

  In other words, *the global minimizer of f is a fixed point of its proximal mapping*.

- It can also be shown that, for closed and *strongly* convex f, $\mathbf{prox}_f(\cdot)$ is always *contractive*, i.e.

$$\|\mathbf{prox}_f(x) - \mathbf{prox}_f(y)\| \leq \kappa\|x - y\|, \quad \text{with} \;\; 0 < \kappa < 1$$

- Such operators play a special role in the *Fixed-Point Theorem*, which we recall next.

Fixed Point Theorem

Let $(\mathcal{X}, \|\cdot\|)$ be a complete normed (aka *Banach*) space (such as, e.g., $\mathbf{R}^n$ with any norm). Also, let $\mathcal{T} : \mathcal{X} \to \mathcal{X}$ be a *contraction*. Then, there is a *unique* x^* such that $\mathcal{T}(x^*) = x^*$.

Moreover, starting with any $x^{(0)} \in \mathcal{X}$, the sequence of iterations

$$x^{(t+1)} = \mathcal{T}(x^{(t)})$$

is guaranteed to converge to x^*, i.e. $\|x^{(t)} - x^*\| \underset{t\to\infty}{\longrightarrow} 0$.

- The theorem suggests the possibility to find a global minimizer of f as a *fixed point* of its associated proximal mapping.
- Specifically, the *Proximal Point Algorith* (PPA) relies on the iterative up-dates performed according to

$$x^{(t+1)} = \mathbf{prox}_{\lambda f}(x^{(t)})$$

- The PPA is guaranteed to converge under rather general conditions on f (that can either be differentiable or sub-differentiable). Particularly, f can be an *extended-value function*!
- When the convexity of f is not strong, its proximal mapping is not a contraction, in general. So, *do we still have a convergence*?
- It turns out, in the case of weakly convex f, the algorithm still converges to their global minimizers (which might no longer be unique).
- Moreover, it converges under the mildest possible assumption, which is simply that a minimizer exists.
- On the practical side, working with proximal mappings is particularly advantageous in view of their special properties which we mention next.

- **Separable sum**

  If f is *separable across two variables*, i.e. $f(x, y) = \varphi(x) + \psi(y)$, then

$$\mathbf{prox}_f(v, w) = \left(\mathbf{prox}_\varphi(v),\, \mathbf{prox}_\psi(w)\right)$$

  More generally, if f is *fully separable*, i.e. $f(x) = \sum_{i=1}^n f_i(x_i)$, then

$$\left(\mathbf{prox}_f(v)\right)_i = \mathbf{prox}_{f_i}(v_i)$$

  for $i = 1, 2, \ldots, n$.

- **Postcomposition**

  If $f(x) = \alpha\,\varphi(x) + b$, with $\alpha > 0$, then

$$\mathbf{prox}_{\lambda f}(v) = \mathbf{prox}_{\alpha\lambda\varphi}(v)$$

- **Precomposition**

  If $f(x) = \varphi(\alpha x + b)$, with $\alpha \neq 0$, then

$$\mathbf{prox}_{\lambda f}(v) = \frac{1}{\alpha}\left(\mathbf{prox}_{\alpha^2\lambda\varphi}(\alpha v + b) - b\right)$$

- **Rotational invariance**

  If $f(x) = \varphi(Qx)$, where $Q^T Q = Q Q^T = I$, then

$$\mathbf{prox}_{\lambda f}(v) = Q^T\, \mathbf{prox}_{\lambda \varphi}(Qv)$$

- **Affine addition**

  If $f(x) = \varphi(x) + a^T x + b$, then

$$\mathbf{prox}_{\lambda f}(v) = \mathbf{prox}_{\lambda \varphi}(v - \lambda\, a)$$

- **Regularization**

  If $f(x) = \varphi(x) + (\rho/2)\|x - a\|_2^2$, then

$$\mathbf{prox}_{\lambda f}(v) = \mathbf{prox}_{\tilde{\lambda} \varphi}\left((\tilde{\lambda}/\lambda)\, v + (\rho \tilde{\lambda})\, a \right)$$

  where $\tilde{\lambda} = \lambda/(1 + \lambda\rho)$.

- Proximal mappings have several properties which make them resemble *orthogonal projections* onto convex sets. To understand these properties, we need to recall the definition of *Legendre transform*.

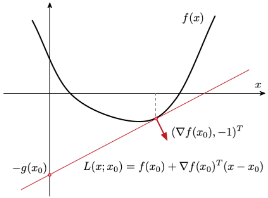
- Let $f : \mathbf{R}^n \to \mathbf{R}$ be convex and differentiable. Then, at each $x_0 \in \mathbf{R}^n$, one can define the supporting (aka *tangent*) line as given by

$$L(x; x_0) = f(x_0) + \nabla f(x_0)^T (x - x_0) \le f(x)$$

for all $x \in \mathbf{R}^n$.

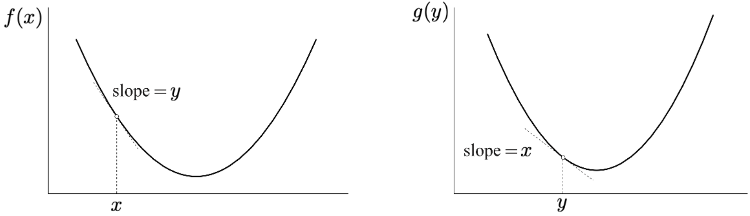
Let $g(x) = -L(0; x)$, so that the tangent line crosses the vertical axis at $-g(x_0)$. Hence, for each x, we have

$$\begin{cases} g(y) = y^T x - f(x) \\ y = \nabla f(x) \end{cases}$$



- These two equations define precisely what the *Legendre transform* (LT) of f is.

- In particular, the LT enables a *dual representation* of $f(x)$ in terms of its *"slopes"* described by the *dual variable* y.



If f admits a supporting line at x with slope y, then g admits a supporting line at y with slope x.

- Moreover, it can be shown that, if f admits a *strict* supporting line at x with slope k, then g admits a tangent supporting line at y with slope

$$\nabla g(y) = x$$

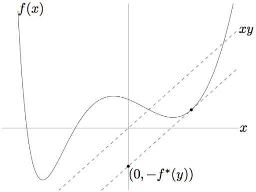
and, hence, g is differentiable.

- The *conjugate transform* (aka the *Legendre-Fenchel transform*) extends the LT to sub-differentiable convex functions, and it is defined as

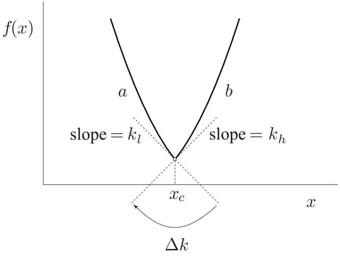
$$f^*(y) = \sup_{x \in \mathbf{dom}\, f} \left(y^T x - f(x) \right)$$

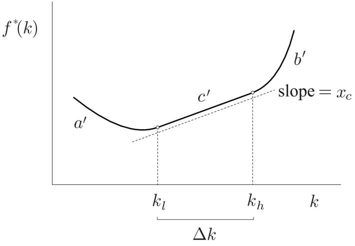
with $\mathbf{dom}\, f^* = \{y \mid y^T x - f(x) \text{ is bounded}\}$.

Important: f^* is a convex function, whether or not f is convex.



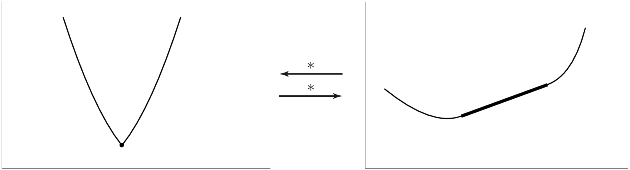
- Similarly to the LT, the conjugate transform "encodes" f in terms of its tangents comprising the *dual space* of variable y.





- Each point $(x, f(x))$ on the differentiable branches of f admits a strict supporting line (or hyperplane) with slope $\nabla f(x) = k$.
- The non-differentiable point x_c admits *infinitely many* supporting lines with slopes in the range $[k_l, k_h]$.
- So, each point of $f^*(k)$ with $k \in [k_l, k_h]$ must admit a supporting line with constant slope x_c (branch c').
- In this case, we say that f^* is *affine* or *linear* over (k_l, k_h).

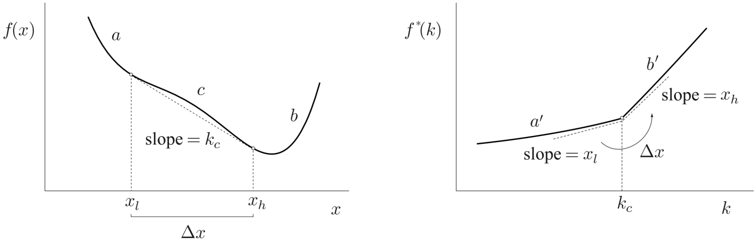
- In the case, we f is convex, we have $f^{**} = f$.



- A convex function f having an affine part has f^* with one non-differentiable point.

- More precisely, if f is affine over (x_l, x_h) with slope k_c in that interval, then f^* will have a non-differentiable point at k_c with left- and right-derivatives at k_c given by x_l and x_h, respectively.

- In the case when f is non-convex, the conjugate transform follows the boundary of the *convex hull* of **epi** f.



- Define the *convex extrapolation* of f to be the function obtained by replacing its non-convex branch (c) by the supporting line that connects the two convex branches of f (a and b).

- Then, the conjugate transforms of f and its convex extrapolation both yield f^*. For this reason, f^{**} is also called the *convex envelope* of f.

- The conjugate transform yields only convex functions, i.e f^* is convex and so is f^{**}.
- Points of f are transformed into slopes of f^*, and slopes of f are transformed into points of f^*.
- Non-differentiable points are transformed into affine branches of f^*.
- Affine or non-convex branches of f are transformed into non-differentiable points of f^* (the only two cases).
- $f(x) = f^{**}(x)$ *if and only if* f admits a supporting hyperplane at x.
- If f^* is differentiable at y, then $f(x) = f^{**}(x)$ at $x = \nabla f^*(y)$.

- Define $f(x) = -\log x$, then

$$f^*(y) = \sup_{x>0} (yx + \log x) = \begin{cases} -1 - \log(-y), & y < 0 \\ \infty, & y \geq 0 \end{cases}$$

- Define $f(x) = (1/2)x^T Q x$, with $x \in \mathbf{R}^n$ and $Q \in \mathbf{S}^n_{++}$. Then,

$$f^*(y) = \sup_x \left(y^T x - (1/2) x^T Q x \right) = (1/2) y^T Q^{-1} y$$

- Define $f(X) = \log\det X^{-1}$, with $X \in \mathbf{S}^n_{++}$. Then,

$$f^*(Y) = \sup_{X \succ 0} \left(\mathbf{tr}(YX) + \log\det X \right) = \log\det(-Y)^{-1} - n,$$

with $\mathbf{dom}\, f^* = -\mathbf{S}^n_{++}$.

- Let $f(x) = I_{\mathcal{C}}(x)$ be the *indicator function* of a set $\mathcal{C} \subset \mathbf{R}^n$. Then,

$$I^*_{\mathcal{C}}(y) = \sup_{x \in \mathcal{C}} y^T x,$$

which is the *support function* of C.

- Let $f(x) = \|x\|$ be a *norm* on $\mathbf{R}^n$ with its associated *dual norm* $\|\cdot\|_*$. Then the conjugate of f is given by

$$f^*(y) = \begin{cases} 0, & \|y\|_* \leq 1 \\ \infty, & \text{otherwise} \end{cases} = I_{\mathcal{B}^*}(y)$$

which is the *indicator function of the unit ball in the dual-norm space*.

- It can also be shown that the conjugate of $f(x) = (1/2)\|x\|^2$ is equal to $f^*(y) = (1/2)\|y\|^2_*$.

- *Fenchel's inequality* extends $(1/2)x^T x + (1/2)y^T y \geq x^T y$ to non-quadratic functions as

$$f(x) + f^*(y) \geq x^T y$$

For example, $(1/2)x^T Qx + (1/2)y^T Q^{-1} y \geq x^T y$ for $Q \in \mathbf{S}^n_{++}$.

- Some other useful properties of f^* are listed below.

$f(x)$	$f^*(y)$
$f_1(x_1) + f_2(x_2)$	$f_1^*(y_1) + f_2^*(y_2)$
$ag(x)$	$ag^*(y/a)$
$g(Ax)$	$f^*(y) = g^*(A^{-T}y)$
$g(x-b)$	$g^*(y) + b^T y$
$g(x) + a^T x + b$	$g^*(y-a) - b$
$\inf_{u+v=x}(f_1(u) + f_2(v))$	$f_1^*(y) + f_2^*(y)$

- Note that the operation $\inf_{u+v=x}(f_1(u) + f_2(v))$ is called the *infimal convolution* of f_1 and f_2.

- Recall the subgradient characterization of **prox** f is given by

$$y = \mathbf{prox}_f(x) \quad \Longleftrightarrow \quad x - y \in \partial f(y)$$

for any convex and sub-differentiable $f : \mathbf{R}^n \to \mathbf{R}$.

- However, by the definition of the conjugate transform, we then have

$$x - y \in \partial f(y) \quad \Longleftrightarrow \quad y \in \partial f^*(x - y) \quad \Longleftrightarrow \quad x - y = \mathbf{prox}_{f^*}(x)$$

- The above relations suggest a very important result, namely

$$x = \mathbf{prox}_f(x) + \mathbf{prox}_{f^*}(x)$$

which is known as *Moreau decomposition* – the main relationship between *proximal operators and duality*.

- Note that, more generally, we have

$$x = \mathbf{prox}_{\lambda f}(x) + \lambda\, \mathbf{prox}_{\lambda^{-1} f^*}(x/\lambda)$$

- Let $\mathbf{V}$ be a linear (hence convex and closed) subspace in $\mathbf{R}^n$. Also, let f be the indicator function of $\mathbf{V}$, i.e. $f(x) = I_{\mathbf{V}}(x)$.

- In this case, $\mathbf{prox}_f(x) = \mathcal{P}_{\mathbf{V}}(x)$, where $\mathcal{P}_{\mathbf{V}} : \mathbf{R}^n \to \mathbf{V}$ denotes the operator of orthogonal projection onto $\mathbf{V}$. Indeed,

$$\mathbf{prox}_f(x) = \arg\min_{x'} \left\{ \frac{1}{2}\|x' - x\|_2^2 + I_{\mathbf{V}}(x') \right\} =$$

$$= \arg\min_{x' \in \mathbf{V}} \left\{ \frac{1}{2}\|x' - x\|_2^2 \right\} = \mathcal{P}_{\mathbf{V}}(x)$$

- It is straightforward to see that the conjugate of $f(x) = I_{\mathbf{V}}(x)$ is equal to $f^*(x) = I_{\mathbf{V}^\perp}(x)$, with $\mathbf{V}^\perp$ being the *orthogonal complement* of $\mathbf{V}$ in $\mathbf{R}^n$, which suggests that $\mathbf{prox}_{f^*}(x) = \mathcal{P}_{\mathbf{V}^\perp}(x)$.

- In this case, Moreau decomposition suggests

$$x = \mathbf{prox}_f(x) + \mathbf{prox}_{f^*}(x) = \mathcal{P}_{\mathbf{V}}(x) + \mathcal{P}_{\mathbf{V}^\perp}(x)$$

which is nothing else but the *orthogonal decomposition* of x w.r.t. $\mathbf{V}$.

- In general, Moreau decomposition gives a simple way to obtain the proximal operator of a function f in terms of the proximal operator of f^*.

- For example, if $f(x) = \|x\|$ is a general norm, then $f^* = I_{\mathcal{B}^*}$, where

$$\mathcal{B}^* = \{x \mid \|x\|_* \leq 1\}$$

  is the unit ball for the dual norm $\|\cdot\|_*$.

- By Moreau decomposition, this implies that

$$x = \mathbf{prox}_f(x) + \mathcal{P}_{\mathcal{B}^*}(x)$$

  thus suggesting that $\mathbf{prox}_f(x) = x - \mathcal{P}_{\mathcal{B}^*}(x)$.

- Thus, we can easily evaluate $\mathbf{prox}_f$ if we know how to project on $\mathcal{B}^*$.

- In general, Moreau decomposition is very useful in cases when computing $\mathbf{prox}_f$ is “expansive", while computing $\mathbf{prox}_{f^*}$ is “cheap" (or *vice versa*).

- **Quadratic function**

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c \quad \Longleftrightarrow \quad \mathbf{prox}_{\lambda f} = (I + \lambda A)^{-1}(x - \lambda b)$$

- **Euclidean norm**

$$f(x) = \|x\|_2 \quad \Longleftrightarrow \quad \mathbf{prox}_{\lambda f} = \begin{cases} (1 - \lambda/\|x\|_2)x, & \text{if } \|x\|_2 \geq \lambda \\ 0, & \text{otherwise} \end{cases}$$

- **Logarithmic barrier**

$$f(x) = -\sum_{i=1}^{n} \log x_i \quad \Longleftrightarrow \quad \left(\mathbf{prox}_{\lambda f}(x)\right)_i = \frac{x_i + \sqrt{x_i^2 + 4\lambda}}{2}, \; i = 1, \ldots, n$$

- **ℓ_1 norm**

$$f(x) = \|x\|_1 \quad \Longleftrightarrow \quad \mathbf{prox}_{\lambda f}(x) = \mathcal{S}_\lambda(x) = (|x| - \lambda)_+ \operatorname{sign}(x)$$

- Consider the following optimization problem

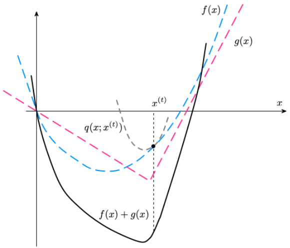
$$\min_x \; f(x) + g(x)$$

where f is convex with a *Lipschitz continuous* gradient (i.e., $\exists L > 0$: $\|\nabla f(x) - \nabla f(y)\|_2^2 \le L\|x - y\|_2^2, \forall x, y$) and g is closed and convex but only sub-differentiable.

Given some sub-optimal point $x^{(t)}$, let $q(x; x^{(t)})$ be a *global over-estimator* (aka *majorizer*) of $f(x)$ defined as

$$q(x; x^{(t)}) = f(x^{(t)}) + \nabla f(x^{(t)})^T (x - x^{(t)}) +$$

$$+ \frac{\kappa}{2}\|x - x^{(t)}\|_2^2, \quad \text{with } \kappa > L.$$



- The majorizer satisfies $q(x^{(t)}; x^{(t)}) = f(x^{(t)})$, while $q(x; x^{(t)}) \ge f(x)$ for all $x \ne x^{(t)}$. Therefore

$$q(x; x^{(t)}) + g(x) \ge f(x) + g(x)$$

- The principal idea of the *Proximal Gradient Method* (PGM) is to minimize $f(x) + g(x)$ based on the *majorization-minimization* iterations of the form
$$x^{(t+1)} = \arg\min_x \{q(x; x^{(t)}) + g(x)\}$$

- In particular, using completion of squares, $q(x; x^{(t)})$ can be expressed as
$$q(x; x^{(t)}) = \frac{\kappa}{2}\left\|x - (x^{(t)} - \frac{1}{\kappa}\nabla f(x^{(t)}))\right\|_2^2 + const$$
where the last term is independent of x. As a result, we have
$$x^{(t+1)} = \arg\min_x \left\{\frac{\kappa}{2}\left\|x - (x^{(t)} - \frac{1}{\kappa}\nabla f(x^{(t)}))\right\|_2^2 + g(x)\right\}$$

- Consequently, the *PGM iterations* are defined by
$$x^{(t+1)} = \mathbf{prox}_{(1/\kappa)g}\left(x^{(t)} - \frac{1}{\kappa}\nabla f(x^{(t)})\right)$$

- In a more general form, the PGM iterations can be also defined by

$$\boxed{x^{(t+1)} = \mathbf{prox}_{\gamma^{(t)} g}\left(x^{(t)} - \gamma^{(t)} \nabla f(x^{(t)})\right)}$$

  for values of the *step-size parameter* $\gamma^{(t)}$ (with $0 < \gamma^{(t)} < 2/L$).

- Note that, if $g(x) = 0$, the PGM iterations are reduced to

$$x^{(t+1)} = x^{(t)} - \gamma^{(t)} \nabla f(x^{(t)})$$

  suggesting that *PGM becomes GDM*.

- On the other hand, if $f(x) = 0$, the PGM iterations are reduced to

$$x^{(t+1)} = \mathbf{prox}_{\gamma^{(t)} g}(x^{(t)})$$

  which suggests that *PGM becomes PPA*.

- When $g(x) = I_{\mathcal{C}}(x)$, with $\mathcal{C} \in \mathbf{R}^n$ being convex and closed, the PGM iterations have the form of

$$x^{(t+1)} = \mathcal{P}_{\mathcal{C}}\left(x^{(t)} - \gamma^{(t)} \nabla f(x^{(t)})\right)$$

  which is also known as the *Projected Gradient Method*.

- Each PGM iteration is based on a *forward-backward splitting* scheme, *viz.*

$$x^{(t+1)} = \underbrace{\mathbf{prox}_{\gamma^{(t)} g}}_{\text{backward step}} \underbrace{\left(x^{(t)} - \gamma^{(t)} \nabla f(x^{(t)})\right)}_{\text{forward step}}$$

- It can be broken up into a *forward* (*explicit*) gradient step using the function f, and a *backward* (*implicit*) step using the function g.

Forward-Backward Algorithm

---

given $x^{(0)}$, $\epsilon \in (0, \min\{1, L^{-1}\})$, $\gamma^{(t)} \in [\epsilon, 2/L - \epsilon]$, $\lambda^{(t)} \in [\epsilon, 1]$

for $t = 0, 1, 2, \ldots$

1. `Set` $y^{(t)} := x^{(t)} - \gamma^{(t)} \nabla f(x^{(t)})$
2. `Update` $x^{(t+1)} := x^{(t)} + \lambda^{(t)} (\mathbf{prox}_{\gamma^{(t)} g}(y^{(t)}) - x^{(t)})$

end

- Note that the above version of PGM incorporates *relaxation parameters* $\{\lambda^{(t)}\}$ (yielding the standard form, if $\lambda^{(t)} = 1$ for all t).

- Let $f(x) = \|Ax - b\|_2^2/2$ and $g(x) = I_{\mathcal{C}}(x)$, with $\mathcal{C} \subset \mathbf{R}^n$ being convex and closed. The resulting minimization problem is

$$\min_{x \in \mathcal{C}} \|Ax - b\|_2^2$$

which is known as *constrained least-squares (LS)*.

- Since $\nabla f : x \mapsto A^T(Ax - b)$ has $L = \|A\|^2 = \sigma_{\max}(A)^2$, the PGM yields the *projected Landweber method*, with its iterations given by

$$x^{(t+1)} = \mathcal{P}_{\mathcal{C}}\left(x^{(t)} + \gamma^{(t)} A^T (b - Ax^{(t)})\right)$$

for some $\gamma^{(t)} \in [\epsilon, 2/\|A\|^2 - \epsilon]$.

- It is also possible to set $\gamma^{(t)}$ adaptively by means of *line search*.

- There are several different versions of PGM.

Constant-Step Forward-Backward Algorithm (FBA)

given $x^{(0)}$, $\epsilon \in (0, 3/4)$, $\lambda^{(t)} \in [\epsilon, 3/2 - \epsilon]$

for $t = 0, 1, 2, \ldots$

1. Set $y^{(t)} := x^{(t)} - (1/L)\nabla f(x^{(t)})$
2. Update $x^{(t+1)} := x^{(t)} + \lambda^{(t)}(\mathbf{prox}_{(1/L)\gamma^{(t)}g}(y^{(t)}) - x^{(t)})$

end

Beck-Teboulle Proximal Gradient Algorithm (BTA)

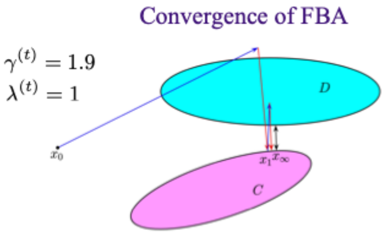
given $x^{(0)}$, $z^{(0)} = x^{(0)}$, $\tau^{(0)} = 1$

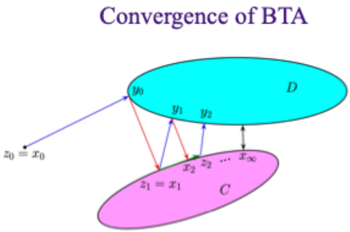
for $t = 0, 1, 2, \ldots$

1. Set $\tau^{(t+1)} := \frac{1+\sqrt{4(\tau^{(t)})^2+1}}{2}$ & $\lambda^{(t)} := 1 + \frac{\tau^{(t)}-1}{\tau^{(t+1)}}$
2. Set $y^{(t)} := z^{(t)} - (1/L)\nabla f(z^{(t)})$
3. Update $x^{(t+1)} := \mathbf{prox}_{(1/L)g}(y^{(t)})$
4. Update $z^{(t+1)} := x^{(t)} + \lambda^{(t)}(x^{(t+1)} - x^{(t)})$

end

- All these methods guarantee convergence to a solution of the original problem, i.e., to an optimal point x^* at which $\bar{0} \in \nabla f(x^*) + \partial g(x^*)$.
- Consider the problem of finding x^* in $\mathcal{C} \in \mathbf{R}^n$ which is at the shortest possible distance $d_{\mathcal{D}}$ from another set $\mathcal{D} \in \mathbf{R}^n$.
- In this case, we formally have $f(x) = d_{\mathcal{D}}(x)^2/2$ and $g(x) = I_{\mathcal{C}}(x)$.





- For the reasons demonstrated by the above example, BTA is often preferred in practice.

- Consider the previous problem

$$\min_x \; f(x) + g(x)$$

where now *both* f and g are sub-differentiable (as well as closed and convex, as before).

- The *Douglas-Rachford (splitting) algorithm* (DRA) iterates according to

$$x^{(t+1)} = \mathbf{prox}_f(y^{(t)})$$
$$y^{(t+1)} = y^{(t)} + \mathbf{prox}_g(2x^{(t+1)} - y^{(t)}) - x^{(t+1)}$$

- This method is useful when f and g have inexpensive proximity mappings.

- It should be noted, however, the DRA is not *symmetric* in the roles of f and g.

- Let $F(y) = y + \mathbf{prox}_g(2\,\mathbf{prox}_f(y) - y) - \mathbf{prox}_f(y)$. Then, the DRA amounts to *fixed-point iterations* of the form

$$y^{(t+1)} = F(y^{(t)})$$

which can be shown to converge to a fixed point $y^* = F(y^*)$ such that $\bar{0} \in \partial f(y^*) + \partial g(y^*)$.

- The DRA can be re-defined in an *equivalent form* given by

$$\begin{aligned} y^{(t+1)} &= \mathbf{prox}_g(x^{(t)} + z^{(t)}) \\ x^{(t+1)} &= \mathbf{prox}_f(y^{(t+1)} - z^{(t)}) \\ z^{(t+1)} &= z^{(t)} + x^{(t+1)} - y^{(t+1)} \end{aligned}$$

starting with some $x^{(0)}$ and $z^{(0)} = \bar{0}$.

- To further accelerate the convergence, the DRA can be subjected to the procedure of *relaxation* as given by

$$y^{(t+1)} = y^{(t)} + \lambda^{(t)}(F(y^{(t)}) - y^{(t)})$$

with $\lambda^{(t)} \in (0, 2)$.

- The regime with $\lambda^{(t)} \in (0, 1)$ is called *under-relaxation*, while the regime with $\lambda^{(t)} \in (1, 2)$ is called *over-relaxation*.

- The DRA can also be expressed in its *dual form* which is derived via the use of Moreau decomposition. In this form, we have

$$x^{(t+1)} = \mathbf{prox}_f(x^{(t)} - z^{(t)})$$
$$z^{(t+1)} = \mathbf{prox}_{g^*}(z^{(t)} + 2x^{(t+1)} - x^{(t)})$$

This form is preferable when computing $\mathbf{prox}_{g^*}$ is less expansive than computing $\mathbf{prox}_g$.

- In multivariate data analysis, *graphical models* (e.g., *Gaussian Markov Random Fields*) provide a way to discover meaningful interactions between random variables.
- Frequently, learning the structure of a graphical model is equivalent to the problem of learning the *zero-pattern* of Σ^{-1}, where $\Sigma \in \mathbf{S}^n_{++}$ is the covariance of modelled variables.
- Formally, the above problem amounts to solving

$$\min_{X \in \mathbf{S}^n_{++}} \left\{ \overbrace{\mathbf{tr}(CX)}^{\langle C, X \rangle} - \log \mathbf{det}\, X + \mu \sum_{i>j} |X_{i,j}| \right\}$$

  for some $C \in \mathbf{S}^n_{+}$ and $\mu > 0$.
- Let us set $f(X) = \mathbf{tr}(CX) - \log \mathbf{det}\, X$ and $g(X) = \mu \sum_{i>j} |X_{i,j}|$.
- $\mathbf{prox}_{\tau f}(U)$ is given by the *positive solution* to

$$C - X^{-1} + \frac{1}{\tau}(X - U) = 0$$

  while $\mathbf{prox}_{\tau f}(\cdot)$ is *soft-thresholding* (with threshold τ).

- Consider the following problem

$$\min_{x \in \mathcal{C}} f(x)$$

with some closed and convex $\mathcal{C} \in \mathbf{R}^n$.

- In this case, the DRA (with $g(x) = I_{\mathcal{C}}(x)$) yields

$$x^{(t+1)} = \mathbf{prox}_{\tau f}(y^{(t)})$$
$$y^{(t+1)} = y^{(t)} + \mathcal{P}_{\mathcal{C}}(2x^{(t+1)} - y^{(t)}) - x^{(t+1)}$$

- Equivalently, using Moreau decomposition, one can obtain the *primal-dual* form of DRA which yields

$$x^{(t+1)} = \mathbf{prox}_{\tau f}(x^{(t)} - z^{(t)})$$
$$z^{(t+1)} = \mathcal{P}_{\mathcal{C}^{\perp}}(z^{(t)} + 2x^{(t+1)} - x^{(t)})$$

Note that $z^{(t)}$ here is a *dual variable* (i.e., subgradient).

- For $x \in \mathbf{R}^n$ and some $A \in \mathbf{R}^{m \times n}$, consider the following problem

$$\min_x \; f(x) + g(Ax)$$

  where both f and g admit **prox** operators.

- Define $\mathcal{C} = \{(u, v) \mid v = Au\}$. Then, the initial problem is equivalent to minimizing $F(u, v) = f(u) + g(v)$ over $\mathcal{C}$, *viz.*

$$\min_{(u,v)\in\mathcal{C}} \; F(u, v)$$

- Due to the *separability* of proximal mapping, we have

$$\mathbf{prox}_{\tau F}(u, v) = \Big(\mathbf{prox}_{\tau f}(u),\, \mathbf{prox}_{\tau g}(v)\Big)$$

- Using *singular value decomposition* of A, it can be shown that $\mathcal{P}_{\mathcal{C}}$ is defined via solution of a system of linear equations to produce

$$\mathcal{P}_{\mathcal{C}}(u, v) = \begin{bmatrix} u \\ v \end{bmatrix} + \underbrace{\begin{bmatrix} A^T \\ -I \end{bmatrix} (I + AA^T)^{-1}}_{A^\sharp} (v - Au)$$

  ($A^\sharp$ can be precomputed.)

1. https://web.stanford.edu/~boyd/papers/pdf/prox_algs.pdf
2. H. Bauschke and J. Borwein, "*On projection algorithms for solving convex feasibility problems*," SIAM Review, 38(3), pp. 367-426, 1996.
3. A. Beck and M. Teboulle, "*A fast iterative shrinkage-thresholding algorithm for linear inverse problems*," SIAM Journal on Imaging Sciences, 2(1), pp. 183-202, 2009.
4. G. Chen and R. Rockafellar, "*Convergence rates in forward-backward splitting*," SIAM Journal on Optimization, 7(2), pp. 421-444, 1997.
5. P. Combettes and J.-C. Pesquet, "*Proximal splitting methods in signal processing*," Fixed-Point Algorithms for Inverse Problems in Science & Engineering, pp. 185-212, 2011.