Introduction au Compressed sensing,
Algorithmes proximaux[1]

Guillaume Lecué[2]

Résumé

1 Introduction

Le problème du Basis Pursuit est une procédure de minimisation d'une fonction convexe sous contrainte affine :

$$\min_{t\in K} f(t) \tag{BP}$$

où $f(t) = \|t\|_1$ où $K = \{t \in \mathbb{R}^N : At = y\}$.

On a vu dans les deux chapitres précédents qu'il était possible de réécrire (BP) comme un problème de programmation linéaire. Il existe aussi des algorithmes qui permettent d'implémenter des problèmes du type (BP) sous des contraintes assez faibles.

Dans ce chapitre nous présentons quelques exemples de ces algorithmes qui peuvent être utilisé dans de nombreux autres problèmes d'optimisation au delà du Basis Pursuit. Cet exposé donne l'occasion de rappeler quelques notions en optimisation convexe, comme le sous-gradient et les opérateurs proximaux.

2 L'algorithme Forward-Backward

Dans le chapitre précédent, nous avons vu que la méthode de descente de gradient est un algorithme itératif de la forme

$$x_{k+1} = x_k - \eta_k \nabla f(x_k)$$

qui a pour but de chercher le minimum d'une fonction convexe f pour un problème d'optimisation sans contrainte.

Dans le cas d'un problème de minimisation sous contraintes, une variante de la méthode de descente de gradient est de chercher des directions de descentes tout en s'assurant que les itérés successifs sont bien dans l'ensemble de contraintes. La méthode s'écrit alors

$$x_{k+1} = x_k + p_k$$

où

$$p_k \in \underset{p}{\operatorname{argmin}} \Big(\big\langle \nabla f(x_k), p \big\rangle + \frac{1}{2\eta_k} \|p\|_2^2 : x_k + p \in K \Big).$$

1. Notes du cours du 26 janvier 2016

2. CNRS, CREST, ENSAE. Bureau E31, 3 avenue Pierre Larousse. 92 245 Malakof. Email : guillaume.lecue@ensae.fr.

La contrainte "$x_k + p \in K$" est ici pour assurer que l'itération suivante x_k reste bien dans K.

Une autre idée est de considérer une itérée de la méthode de descente de gradient comme si le problème n'était pas contraint et, ensuite, de projeter cette itération sur la contrainte. L'algorithme s'écrit alors :

1. étape de descente :
$$y_{k+1} = x_k - \eta_k \nabla f(x_k)$$
2. étape de projection :
$$x_{k+1} = \mathrm{proj}_K(y_{k+1})$$
où $\mathrm{proj}_K(\cdot)$ est l'opérateur de projection sur K défini par $\mathrm{proj}_K(y) \in \mathrm{argmin}_{z\in K} \|y-z\|_2$.

Cet algorithme est appelée algorithme du gradient projeté.

Pour les deux algorithmes précédant la fonction objectif doit être différentiable. Ce n'est pas le cas du Basis Pursuit. On va alors considérer d'autres procédures. Mais avant cela on va introduire les notions de sous-gradient et de fonction proximale.

Une manière de voir la méthode du gradient projeté est d'écrire

$$x_{k+1} \in \underset{u\in K}{\mathrm{argmin}} \|u - (x_k - \eta_k \nabla f(x_k))\|_2 = \underset{u\in \mathbb{R}^N}{\mathrm{argmin}} \Big(\frac{1}{2} \|u - (x_k - \eta_k \nabla f(x_k))\|_2^2 + i_K(u) \Big)$$

où $i_K(u)$ est la fonction indicatrice de K définie par

$$i_K(u) = \begin{cases} +\infty & \text{quand } u \notin K \\ 0 & \text{quand } u \in K. \end{cases}$$

Ainsi, si on définit la fonction

$$\mathrm{prox}_{i_K}(z) \in \underset{u}{\mathrm{argmin}} \Big(\frac{1}{2} \|u - z\|_2^2 + i_K(u) \Big) \tag{2.1}$$

on peut réécrire l'algorithme du gradient projeté comme

$$x_{k+1} = \mathrm{prox}_{i_K}(x_k - \eta_k \nabla f(x_k)).$$

On peut étendre la définition de l'opérateur introduit dans 2.1 à toute fonction convexe.

Définition 2.1. *Soit $f : \mathbb{R}^N \to \mathbb{R}$ une fonction convexe. L'**opérateur proximal de** f est la fonction définie pour tout $z \in \mathbb{R}^N$ par*

$$\mathrm{prox}_f(z) \in \underset{u}{\mathrm{argmin}} \Big(\frac{1}{2} \|u - z\|_2^2 + f(u) \Big).$$

Exemple 2.2. *[**Fonction proximale de la norme** ℓ_1] L'opérateur proximal de $\ell_1 : x \to \|x\|_1$ est l'opérateur de seuillage doux. En effet, on voit que la fonction à optimiser définissant l'opérateur proximal de $\gamma\ell_1$ (pour $\gamma > 0$) est décomposable : pour tout $u \in \mathbb{R}^N$,*

$$\frac{1}{2} \|u - z\|_2^2 + \gamma \|u\|_1 = \sum_{j=1}^{N} \Big(\frac{1}{2}(u_j - z_j)^2 + \gamma |u_j| \Big).$$

Il suffit alors de minimiser chaque terme de cette somme. On est donc amené à trouver une solution au problème (à une variable) : pour tout $s \in \mathbb{R}$,

$$\underset{t\in\mathbb{R}}{\mathrm{argmin}} \Big(\frac{1}{2}(s-t)^2 + \gamma|t| \Big).$$

Pour obtenir ce minimum, on étudie $\varphi(t) = (s-t)^2/2 + \gamma|t|$ sur $(-\infty, 0)$, $\{0\}$ et $(0, +\infty)$. On voit que φ atteint son minimum en $s-\gamma$ sur $(0,\infty)$ quand $s-\gamma \geq 0$ et que ce minimum est plus petit que $\varphi(0)$ sous cette condition. De même, quand $s \leq -\gamma$, φ atteint son minimum en $s+\gamma$ sur $(-\infty, 0)$ et est plus petit qu'en zéro. Finalement, quand $-\gamma \leq s \leq \gamma$, on voit que le minimum de φ est atteint en 0. On a donc

$$\operatorname*{argmin}_{t\in\mathbb{R}} \Big(\frac{1}{2}(s-t)^2 + \gamma|t|\Big) = \begin{cases} s-\gamma & si\ s \geq \gamma \\ 0 & si\ -\gamma \leq s \leq \gamma \\ s+\gamma & si\ s \leq -\gamma \end{cases} = \operatorname{sgn}(s)\big(|s|-\gamma\big)_+$$

qui est l'opérateur de seuillage doux pour un seuil égal à γ. L'opérateur de seuillage dur étant $s \to sI(|s| \geq \gamma)$ qui peut être obtenu comme l'opérateur proximal de l'indicatrice de 0 :

$$\operatorname*{argmin}_{t\in\mathbb{R}} \Big(\frac{1}{2}(s-t)^2 + \frac{\gamma^2}{2} I(t \neq 0)\Big) = sI(|s| \geq \gamma)$$

où $I(t \neq 0)$ vaut 1 quand $t \neq 0$ et 0 sinon. On a donc pour tout $z \in \mathbb{R}^N$,

$$\boxed{\operatorname{prox}_{\gamma\|\cdot\|_1}(z) = \big(\operatorname{sgn}(z_j)(|z_j| - \gamma)_+\big)_{j=1}^N} \tag{2.2}$$

qui est l'opérateur de seuillage doux par coordonnées et

$$\boxed{\operatorname{prox}_{(\gamma^2/2)\|\cdot\|_0}(z) = \big(z_j I(|z_j| \geq \gamma)\big)_{j=1}^N} \tag{2.3}$$

qui est l'opérateur de seuillage dur par coordonnées.

On peut généraliser la méthode du gradient projeté à n'importe quel problème de la forme

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

où f et g sont convexes et f est, de plus, différentiable. On obtient alors la méthode du gradient proximal aussi appelé <u>forward backward splitting algorithm (FBS)</u> :

$$\boxed{x_{k+1} = \operatorname{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k))}$$

où $(\eta_k)_k$ est la suite donnée des *step sizes.*

Il est parfois utile de voir les algorithmes itératifs comme des algorithmes de point fixe. L'idée est que si x^* est solution d'un problème si et seulement x^* est solution d'un problème de point fixe du type $x = F(x)$ alors une manière naturelle d'approcher x^* est de considérer les itérations $x_{k+1} = F(x_k)$.

On va donc réécrire FBS comme un algorithme de point fixe. Cela expliquera pourquoi le *step size* η_k dans la phase *forward* (explicite) de la descente de gradient est pris ici égal au paramètre η_k apparaissant dans la phase *backward* (implicite) de la "projection" par l'opérateur proximal "$\operatorname{prox}_{\gamma_k g}$". Pour cela, on introduit une notion qui généralise la notion de dérivée dans le cas des fonctions convexes.

Définition 2.3. *Soit $f : \mathbb{R}^N \to \mathbb{R}$ une fonction convexe. On définit la **sous-différentielle de** f **en** x par*

$$\partial^- f(x) = \big\{g \in \mathbb{R}^N : f(x+h) \geq f(x) + \big\langle g, h\big\rangle \ pour\ tout\ h \in \mathbb{R}^N\big\}.$$

*Les éléments de la sous-différentielle sont appelés **sous-gradients**.*

On peut voir que si f est une fonction convexe différentiable en x alors

$$\partial^- f(x) = \{\nabla f(x)\}$$

càd, la sous-différentielle de f en x est réduite au singleton $\nabla f(x)$. En effet, la condition du premier ordre[1] : pour tout h

$$f(x+h) \geq f(x) + \big\langle \nabla f(x), h \big\rangle$$

montre que $\nabla f(x)$ est un sous-gradient de f en x. De plus, pour tout $h \in \mathbb{R}^N$, on a quand $\epsilon \downarrow 0$,

$$\frac{f(x+\epsilon h) - f(x)}{\epsilon} \to \big\langle \nabla f(x), h \big\rangle.$$

En particulier, si $g \in \mathbb{R}^N$ est un sous-gradient de f en x, on aura par passage à la limite quand $\epsilon \downarrow 0$ dans

$$\frac{f(x+\epsilon h) - f(x)}{\epsilon} \geq \big\langle g, h \big\rangle$$

que $\big\langle \nabla f(x), h \big\rangle \geq \big\langle g, h \big\rangle$ et ceci pour tout h donc $g = \nabla f(x)$. Autrement dit $\nabla f(x)$ est l'unique sous-gradient de f en x quand f est différentiable en x.

On a aussi les propriétés suivantes pour toutes fonctions convexes f et g et tout $\gamma \geq 0$:

$$\partial^-(\gamma f)(x) = \gamma \partial^- f(x) \text{ et } \partial^-(f+g)(x) = \partial^- f(x) + \partial^- g(x).$$

Exemple 2.4. *[**Sous-différentielle de l'indicatrice d'un espace affine**] Soit $A \in \mathbb{R}^{m\times N}$ et $y \in \mathbb{R}^m$. On considère l'espace affine des solutions $K = \{x \in \mathbb{R}^N : Ax = y\}$ et l'indicatrice i_K de K. On a pour tout $x \in \mathbb{R}^N$*

$$\boxed{\partial^-(i_K)(x) = \begin{cases} \big(\mathrm{Im}(A^\top)\big)^\perp & \textit{si } x \in K \\ \emptyset & \textit{si } x \notin K \end{cases}}$$

En effet, si $x \in \mathbb{R}^N$ on aura $i_K(x+h) = +\infty$ dès que $x+h \notin K$ et donc $i_K(x+h) \geq i_K(x) + \big\langle g, h \big\rangle$ pour tout $g \in \mathbb{R}^N$. Si $x+h \in K$ alors $i_K(x+h) = 1$ et donc $\partial^-(i_K)(x) = \emptyset$ si $i_K(x) = +\infty$ càd quand $x \notin K$ sinon $g \in \partial^-(i_K)(x)$ si et seulement si $0 \geq \big\langle g, h \big\rangle$ pour tout h tel que $x+h \in K$. Or quand $x \in K$, on a $x+h \in K$ si et seulement si $Ah = 0$. On a donc, pour tout $x \in K$,

$$\partial^-(i_K)(x) = \{g : \big\langle g, h \big\rangle \leq 0 \textit{ pour tout } h \in \mathrm{Ker}(A)\}.$$

On conclut avec $\mathrm{Ker}(A) = \big(\mathrm{Im}(A^\top)\big)^\perp$.

Exemple 2.5. *[**Sous-différentielle d'une norme**] Soit $\|\cdot\|$ une norme sur $\mathbb{R}^N$, en particulier, c'est une fonction convexe admettant donc une sous-différentielle. Pour calculer la sous-différentielle de $\|\cdot\|$, il est utile d'introduire la norme duale de $\|\cdot\|$: pour tout $g \in \mathbb{R}^N$,*

$$\|z\|_* = \sup_{\|x\|\leq 1} \big\langle z, x \big\rangle.$$

On note par B_* la boule unité de la norme duale et par S_* sa sphère unité. On a

$$\boxed{\partial^- \|\cdot\|(x) = \big\{g \in S_* : \big\langle g, x \big\rangle = \|x\| \big\}.}$$

1. On obtient cette condition en remarquant que pour une fonction convexe f, $f(x+th) \leq tf(x+h) + (1-t)f(x)$ pour tout $0 < t \leq 1$ et donc $f(x+h) \geq f(x) + (g(t) - g(0))/t$ où $g(t) = f(x+th)$ et $g'(0) = \big\langle \nabla f(x), h \big\rangle$. Alors, en passant à la limite quand $t \downarrow 0$, on obtient la propriété de premier ordre.

En effet, si $g \in S_*$ est normant pour x alors pour tout $h \in \mathbb{R}^N$,

$$\|x+h\| - \|x\| \geq \langle g, x+h \rangle - \langle g, x \rangle = \langle g, h \rangle$$

donc $g \in \partial^- \|\cdot\|(x)$. Réciproquement, soit $g \in \mathbb{R}^N$ tel que pour tout $h \in \mathbb{R}^N$

$$\|x+h\| - \|x\| \geq \langle g, h \rangle.$$

En particulier, $\|h\| \geq \langle g, h \rangle$ pour tout h et donc $\|g\|_* \leq 1$. De plus pour $h = -x$, on a $\langle g, x \rangle \geq \|x\|$ donc $\|g\|_* \geq 1$. On en déduit que $\|g\|_* = 1$ et que $\|x\| = \langle g, x \rangle$.

Un exemple classique est celui de la norme ℓ_1, pour laquelle on a

$$\partial^- \|\cdot\|_1(x) = \left\{ g \in S_\infty^{N-1} : g_i = \operatorname{sgn}(x_i), \forall i \in I \right\} = \{ g \in \mathbb{R}^N : g_i = \operatorname{sgn}(x_i), \forall i \in I \text{ et } |g_i| \leq 1, \forall i \notin I \}$$

où $I = \operatorname{supp}(x)$.

Proposition 2.6. *Soit $f : \mathbb{R}^N \to \mathbb{R}$ une fonction convexe. Il y a équivalence entre :*

1. *x^* est solution de $\min_x f(x)$*
2. *$0 \in \partial^- f(x^*)$*

Démonstration. Si x^* minimise f sur $\mathbb{R}^N$ alors pour tout $h \in \mathbb{R}^N$, $f(x+h) \geq f(x)$ et donc 0 est une sous-gradient de f en x.

Réciproquement, si 0 est un sous-gradient de f en x^* alors par définition, pour tout h, on a $f(x^*+h) \geq f(x^*)$ et donc f atteint bien son minimum sur $\mathbb{R}^N$ en x^*. ∎

Proposition 2.7. *Soit $f : \mathbb{R}^N \to \mathbb{R}$ une fonction convexe différentiable et $g : \mathbb{R}^N \to \mathbb{R}$ une fonction convexe. Soit $\gamma > 0$, il y a équivalence entre :*

1. *x^* est solution du problème $\min_x f(x) + g(x)$*
2. *$x^* = \operatorname{prox}_{\gamma g}(x^* - \gamma \nabla f(x^*))$.*

Démonstration. On remarque que pour tout $z \in \mathbb{R}^N$, on a

$$u^* = \operatorname{prox}_{\gamma g}(z) \Leftrightarrow z - u^* \in \gamma \partial^- g(u^*). \tag{2.4}$$

En effet, d'après la Proposition 2.7, u^* est solution du problème proximal associé à γg si et seulement si 0 est un sous-gradient de $u \to \|u-z\|_2^2/2 + \gamma g(u)$ qui est équivalent à $z - u^* \in \gamma \partial^- g(u^*)$.

On en déduit la suite d'équivalences ci-dessous :

$$\begin{aligned} x^* = \operatorname{prox}_{\gamma g}(x^* - \gamma \nabla f(x^*)) &\Leftrightarrow x^* - \gamma \nabla f(x^*) - x^* \in \gamma \partial^- g(x^*) \\ &\Leftrightarrow -\gamma \nabla f(x^*) \in \gamma \partial^- g(x^*) \\ &\Leftrightarrow 0 \in \partial^-(f+g)(x^*) \\ &\Leftrightarrow x \text{ est solution au problème } \min_x f(x) + g(x). \end{aligned}$$

∎

On peut donc maintenant réécrire l'algorithme FBS comme un algorithme de point fixe $x_{k+1} = F(x_k)$ pour la fonction $F(x) = \operatorname{prox}_{\gamma g}(x - \gamma \nabla f(x))$.

Exemple 2.8. ***[FBS pour le LASSO]*** *L'algorithme FBS peut être utilisé pour implémenter le LASSO. On rappelle que le LASSO est solution au problème d'optimisation :*

$$\hat{x} \in \underset{x\in\mathbb{R}^N}{\operatorname{argmin}} \Big(\frac{1}{2} \|y - Ax\|_2^2 + \lambda \|x\|_1 \Big).$$

On pose $f(x) = \|y - Ax\|_2^2/2$, qui est convexe et différentiable et $g(x) = \lambda \|x\|_1$ qui est convexe. L'algorithme FBS s'écrit alors

$$x_{k+1} = \operatorname{prox}_{\gamma_k\lambda\|\cdot\|_1}\big(x_k + \gamma_k A^\top(y - Ax_k)\big)$$

où (γ_k) est une suite donnée de step sizes. L'algorithme $\operatorname{prox}_{\gamma_k\lambda\|\cdot\|_1}$ a été identifié précédemment comme étant l'opérateur de seuillage doux de seuil $\lambda\gamma_k$.

Dans l'exemple précédent, si la régularisation ℓ_1 est remplacée par la régularisation ℓ_0, l'algorithme FBS associé est

$$x_{k+1} = \operatorname{prox}_{\gamma_k\lambda\|\cdot\|_0}\big(x_k + \gamma_k A^\top(y - Ax_k)\big)$$

où (γ_k) est une suite donnée de *step sizes* et l'opérateur proximal $\operatorname{prox}_{\gamma_k\lambda\|\cdot\|_0}$ est l'opérateur de seuillage dur de seuil $\sqrt{2\lambda\gamma_k}$. Dans le cadre du Compressed Sensing, cet algorithme (pour un choix particulier de step-size $\eta_k = 1$ et de paramètre de régularisation λ) a été introduit sous le nom d'*Iterative Hard Thresholding (IHT)*.

Définition 2.9. *Soit $1 \leq s \leq m$ un entier et $x_0 \in \mathbb{R}^N$ un vecteur s-sparse. L'Iterative Hard Thresholding (IHT) est la suite des itération $(x_k)_k$ de point initial x_0 et dont les itérés sont obtenus par*

$$x_{k+1} = H_s\big(x_k + A^\top(y - Ax_k)\big)$$

où $H_s : \mathbb{R}^N \to \mathbb{R}^N$ est l'opérateur ne gardant que les s coefficients ayant les plus grandes valeurs absolues.

On peut aussi voir IHT comme un algorithme de descente de gradient projeté sur l'ensemble non-convexe Σ_s : $x_{k+1} = \operatorname{proj}_{\Sigma_s}(x_k + A^\top(y - Ax_k))$.

L'étude de la convergence de IHT peut se faire sous la condition $RIP(3s)$ pour une constante d'isométrie assez petite.

Théorème 2.10. *Soit $A \in \mathbb{R}^{m\times N}$ telle que pour tout $x \in \Sigma_{3s}$*

$$(1 - \delta) \|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta) \|x\|_2^2 \tag{2.5}$$

pour un certain $0 < \delta < 1$ et un certain $s \in [1, m]$. Soit $x \in \Sigma_s$ et $y = Ax$. On définit la suite des itérations de IHT par un point initial $x_0 \in \Sigma_s$ et la récurrence $x_{k+1} = H_s(x_k + A^\top(y - Ax_k))$. On a pour tout entier k

$$\|x_{k+1} - x\|_2 \leq (2\delta)^k \|x_0 - x\|_2$$

et donc $(x_k)_k$ converge exponentiellement vite vers x dès que $\delta < 1/2$.

Pour démontrer ce résultat, on utilisera le lemme suivant.

Lemme 2.11. *Pour tout vecteur $u, v \in \mathbb{R}^N$ tels que $|\operatorname{supp}(u) \cup \operatorname{supp}(v)| \leq 3s$, on a*

$$|\langle u, (I - A^\top A)v\rangle| \leq \delta \|u\|_2 \|v\|_2 .$$

Démonstration. On note $T = \operatorname{supp}(u) \cup \operatorname{supp}(v)$ et par A_T la matrice de taille $m \times |T|$ qui est extraite de A où seules les colonnes de A dont les indices sont dans T ont été gardées. On suppose que $|T| \leq 3s$. Comme $I - A_T^\top A_T$ est symétrique, on a (voir Lemme 3.8 de la démonstration de RIP pour les matrices sous-gaussiennes)

$$\left\| I - A_T^\top A_T \right\|_{2\to 2} = \sup_{\|x\|_2=1} |\langle x, (I - A_T^\top A_T)x \rangle| = \sup_{\|x\|_2=1} \left| \|x\|_2^2 - \|A_T x\|_2^2 \right| \leq \delta.$$

On en déduit que

$$\begin{aligned} |\langle u, (I - A^\top A)v \rangle| &= |\langle u, v \rangle - \langle Au, Av \rangle| = |\langle u_T, v_T \rangle - \langle A_T u_T, A_T v_T \rangle| \\ &= |\langle u_T, (I - A_T^\top A_T)v_T \rangle| \leq \|u_T\|_2 \left\| (I - A_T^\top A_T)v_T \right\|_2 \\ &\leq \|u_T\|_2 \left\| I - A_T^\top A_T \right\|_{2\to 2} \|v_T\|_2 \leq \delta \|u\|_2 \|v\|_2 . \end{aligned}$$

■

Preuve du Théorème 2.10. On note

$$u_k := x_k + A^\top (y - Ax_k) = x_k + A^\top A(x - x_k). \tag{2.6}$$

Par définition, x_{k+1} est la meilleure approximation s-sparse de u_k, en particulier, meilleure que x ; on a donc

$$\|x_{k+1} - u_k\|_2 \leq \|x - u_k\|_2 .$$

En écrivant

$$\|x_{k+1} - u_k\|_2^2 = \|x_{k+1} - x + x - u_k\|_2^2 = \|x_{k+1} - x\|_2^2 + \|x - u_k\|_2^2 + 2\langle x_{k+1} - x, x - u_k \rangle$$

dans l'inégalité précédente, on obtient

$$\|x - x_{k+1}\|_2^2 \leq 2\langle x_{k+1} - x, u_k - x \rangle = 2\langle x_{k+1} - x, (I - A^\top A)(x_k - x) \rangle.$$

Comme $|\operatorname{supp}(x_{k+1} - x) \cup \operatorname{supp}(x_k - x)| \leq 3s$, on applique le Lemme 2.11 pour obtenir

$$|\langle x_{k+1} - x, (I - A^\top A)(x_k - x) \rangle| \leq \delta \|x_{k+1} - x\|_2 \|x_k - x\|_2 .$$

En combinant les deux dernière inégalités, on obtient bien le résultat annoncé. ■

<u>Conclusion</u> : Pour le problème de minimisation $\min_x f(x) + g(x)$ *où* f *et* g *sont convexes et* f *est différentiable, on peut utiliser l'algorithme FBS pour approcher une solution. Cette méthode est à l'origine de l'introduction de l'algorithme IHT en Compressed Sensing qui converge exponentiellement vite vers la solution du basis pursuit sous une condition RIP.*

3 Douglas-Racheford

La procédure du Basis Pursuit peut être réécrite sous la forme

$$\min_x f(x) + g(x) \text{ où } f(x) = \|x\|_1 \text{ et } g(x) = i_K(x) \tag{3.1}$$

pour $K = \{x \in \mathbb{R}^N : Ax = y\}$, l'espace des solutions. Les deux fonctions f et g sont bien convexes mais aucune n'est différentiable. On ne peut donc pas utiliser une algorithme FBS. On va construire un algorithme qui ne nécessite que la convexité pour être défini.

Définition 3.1. *Etant donné un point initial x_0 et un paramètre $\gamma > 0$, l'**algorithme de Douglas-Rachford** est la suite des itérations $(x_k)_k$ définie par x_0 et la relation de récurrence*

$$\begin{cases} y_{k+1} & = \mathrm{prox}_{\gamma f}(x_k) \\ z_{k+1} & = \mathrm{prox}_{\gamma g}(2y_{k+1} - x_k) \\ x_{k+1} & = x_k + z_{k+1} - y_{k+1} \end{cases}$$

Cette algorithme peut être vu comme un algorithme de point fixe.

Proposition 3.2. *Soit $F(x) = x + \mathrm{prox}_{\gamma f}(2\mathrm{prox}_{\gamma g}(x) - x) - \mathrm{prox}_{\gamma g}(x)$. Si $z = F(z)$ et $x^* = \mathrm{prox}_{\gamma g}(z)$ alors x^* est solution du problème $\min_x f(x) + g(x)$.*

Démonstration. On suppose que $z = F(z)$ et $x^* = \mathrm{prox}_{\gamma g}(z)$. On a alors

$$\mathrm{prox}_{\gamma f}(2x^* - z) = x^* = \mathrm{prox}_{\gamma g}(z).$$

Ainsi, en utilisant l'argument de (2.4), on a

$$-x^* + z \in \gamma\partial^- f(x^*) \text{ et } z - x^* \in \gamma\partial^- g(x^*).$$

On en déduit alors que $0 \in \partial^- f(x^*) + \partial^- g(x^*)$ et donc x^* est bien solution de $\min_x f(x) + g(x)$. ■