Etude d'algorithmes d'optimisation de fonctions convexes dérivables ou non dérivables et application

Bassem JABER - Rémi MARSAL

Juin 2019

Table des matières

1 Méthodes du gradient et illustrations

L'algorithme du gradient est un algorithme d'optimisation de fonctions convexes dérivables définies sur un espace hilbertien et à valeurs dans $\Re$. L'algorithme procède itérativement en améliorant l'approximation qu'il fait du minimum de la fonction à chaque étape. Au point courant, un déplacement est effectué dans la direction opposée au gradient, faisant décroître la fonction. Les différentes variantes de l'algorithme du gradient diffèrent dans la façon de choisir le pas, c'est-à-dire la longueur du déplacement à chaque itération.

1.1 Notions et résulats utilisés

1. Fonctions gradients lipschitz : Une fonction f est gradient lipschitz si et seulement si $\exists\ M > 0$ tel que $\forall\ (x, y) \in \mathrm{dom} f,\ \|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|$
2. Fonctions fortement convexe : Une fonction f est fortement convexe sur un ensemble U si $\exists\, \alpha > 0$ tel que $\forall\, (x, y) \in U^2, \forall\, t \in [0, 1]$, $f(ty+(1-t)x) \leq tf(y)+(1-t)f(x)-\alpha t(1-t)\|x-y\|^2$

1.2 Hypothèses et résultats

Descente de gradient à pas fixe :

Dans le cas de la descente de gradient à pas fixe. La valeur du pas est initialisée et reste constante tout le long de l'algorithme.

1. **Premier cas : fonctions dérivables lipschitziennes**
   Hypothèses :
   $f : \Re^d \longrightarrow \Re$ est une fonction continûment dérivable, convexe et de gradient lipschitz donc $\exists\ M > 0$ vérifiant $\|\nabla f(x)\| \leq M$. Le pas est choisi tel que $t \in]0, \frac{1}{M}]$.

   Résultat :
   L'algorithme converge vers p^* en $\mathrm{O}(\frac{1}{k})$.

2. **Second cas : fonctions dérivables lipschitziennes et fortement convexes**
   Hypothèses :
   $f : \Re^d \longrightarrow \Re$ est une fonction deux fois continûment dérivable et fortement convexe de paramètres $0 < m \leq \|\nabla f(x)\| \leq M$. Le pas est choisi tel que $t \in]0, \frac{2}{m+M}]$.

   Résultat :
   L'algorithme converge vers p^* en $O(c_{PF}^k)$ avec $c_{PF} = 1 - \frac{2tmM}{m+M}$.
   On minimise c_{PF} en choisissant le pas fixe optimal $t_{opt} = \frac{2}{m+M}$, on obtient alors $c_{PF}^{opt} = 1 - \frac{4mM}{(m+M)^2}$.

Descente de gradient utilisant le backtracking :

Cet algorithme choisit à chaque itération une valeur approchée du pas qui minimise la valeur de la fonction. C'est-à-dire qu'il permet d'obtenir une valeur approchée du problème $\min_{t\in\Re} f(x-t\nabla x)$. Son intérêt réside dans le fait qu'il n'est pas nécessaire de connaître les caractéristiques de la fonction à minimiser telles que sa constante de Lipschitz ou son paramètre de forte convexité.

Hypothèses :
$f : \Re^d \longrightarrow \Re$ est une fonction deux fois continûment dérivable et fortement convexe de paramètres $0 < m \leq \|\nabla f(x)\| \leq M$.

Résultat :
L'algorithme converge vers p^* en $O(c_{BT}^k)$ où $0 < c_{BT} = 1 - \frac{2m\alpha}{M}\min(M, \beta) < 1$ avec $0 < \alpha \leq 0,5$ et $0 < \beta \leq 1$.

Descente de gradient utilisant un pas optimal :

À chaque itération, cet algorithme choisit un pas qui minimise la fonction de départ. C'est-à-dire qu'à chaque itération, le pas t^* est choisi de façon à résoudre le problème $\min_{t\in\Re} f(x - t\nabla x)$.

Hypothèses :
$f : \Re^d \longrightarrow \Re$ est une fonction deux fois continûment dérivable et fortement convexe de paramètres $0 < m \leq \|\nabla f(x)\| \leq M$.

Résultat :
L'algorithme converge vers p^* en $O(c_{PO}^k)$ où $0 < c_{PO} = 1 - \frac{2m}{M+m} < 1$.

1.3 Description des algorithmes des variantes de descente de gradient

On choisit $\varepsilon > 0$, la condition d'arrêt et on initialise x à x_0 pour chacun des algorithmes.

Descente de gradient à pas fixe :

On choisit $t \leq \frac{1}{L}$ où L vaut M ou $\frac{m+M}{2}$ selon si la fonction à minimiser est fortement convexe ou juste gradient Lipschitz et $\varepsilon > 0$, la condition d'arrêt.

Algorithm 1 Descente de gradient à pas fixe

while $\|\nabla f(x)\|_2^2 > \varepsilon$ **do**
  $x \leftarrow x - t\nabla f(x)$
end while

Descente de gradient utilisant le backtracking :

On choisit les constantes suivantes $0 < \alpha \leq 0,5$ et $0 < \beta \leq 1$. La seule connaissance de la dérivée de la fonction à minimiser nous suffit pour utiliser cet algorithme.

Algorithm 2 Descente de gradient utilisant le backtracking

while $\|\nabla f(x)\|_2^2 > \varepsilon$ **do**
  $\Delta x \leftarrow \nabla f(x)$
  $t \leftarrow 1$
  while $f(x - \nabla f(x)) > f(x) + \alpha t \|\nabla f(x)\|_2^2$ **do**
    $t \leftarrow \beta t$
  end while
  $x \leftarrow x - t\Delta x$
end while

Descente de gradient utilisant un pas optimal :

Pour appliquer cet algorithme, nous avons besoin de pouvoir résoudre explicitement le problème d'optimisation $\underset{t>0}{\operatorname{argmin}}\ f(x - t\Delta x)$.

Algorithm 3 Descente de gradient utilisant un pas optimal

while $\|\nabla f(x)\|_2^2 > \varepsilon$ **do**
  $\Delta(x) \leftarrow \nabla f(x)$
  $t \leftarrow \underset{t\geq 0}{\operatorname{argmin}}\ f(x - t\Delta x)$
  $x \leftarrow x - t\Delta x$
end while

1.4 Comparaison des performances des algorithmes

Avec une convergence en $O(\frac{1}{k})$, l'algorithme de descente de gradient à pas fixe dans le cas où f est continûment dérivable, convexe et de gradient lipschitz, est le moins performant.

Lorsque la fonction f à minimiser est fortement convexe, les algorithmes de descente de gradient ont tous une convergence en $O(c^k)$ avec $0 < c < 1$. En comparant les valeurs des différentes constantes c, on peut alors comparer les vitesses de convergence qui sont les bornes supérieures des limites théoriques des trois algorithmes présentés ci-dessus.

On rappelle les différentes constantes de convergence :

— Descente de gradient à pas fixe optimal : $c_{PF}^{opt} = 1 - \frac{4mM}{(m+M)^2}$

— Descente de gradient backtracking : $c_{BT} = 1 - \frac{2m\alpha}{M} \min(M, \beta)$ avec $0 < \alpha \leq 0,5$ et $0 < \beta \leq 1$

— Descente de gradient à pas optimal : $c_{opt} = 1 - \frac{2m}{m+M}$

$$c_{opt} - c_{BT} = 2m\frac{\alpha \min(M,\beta)(m+M) - M}{M(m+M)} \leq 0 \text{ pour } m << M \text{ (cas général)}$$

$$c_{PF}^{opt} - c_{opt} = 2m\frac{m-M}{m+M} \leq 0$$

Finalement on peut classer les constantes de convergence ainsi : $c_{PF}^{opt} \leq c_{opt} \leq c_{BT}$.

On peut noter que lorsqu'on a affaire à des fonctions gradients Lipschitz et fortement convexes, l'algorithme à privilégier est l'algorithme de descente de gradient à pas fixe optimal. Si l'on ne connait pas la constante la constante de Lipschitz ou le paramètre de forte convexité, il faut alors choisir l'algorithme à pas optimal. Enfin si résoudre le problème d'optimisation permettant de calculer le pas optimal n'est pas possible ou prend trop de temps, il faut implémenter l'algorithme du gradient backtracking.

1.5 Preuves de convergence des algorithmes

Descente de gradient à pas fixe :

1. **Premier cas :**
   On a, par hypothèse sur $f(x), \forall (x,y) \in \Re^{2d}$:

$$f(x) + \langle \nabla f(x), y - x \rangle \leq f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2$$

   Alors, $\forall n \in \mathbb{N}$:

$$f(x_{n+1}) \leq f(x_n) + \langle \nabla f(x_n), x_{n+1} - x_n \rangle + \frac{L}{2}\|y - x\|^2$$

Or :

$$x_{n+1} - x_n = -t\nabla f(x_n)$$

Soit :

$$f(x_{n+1}) \leq f(x_n) - t\|\nabla f(x)\|^2 + \frac{L}{2}t^2\|\nabla f(x)\|^2$$

$$f(x_{n+1}) \leq f(x_n) - t(1 - \frac{tL}{2})\|\nabla f(x)\|^2$$

Or $\eta L \leq 1$.
Donc :

$$f(x_{n+1}) \leq f(x_n) - \frac{t}{2}\|\nabla f(x)\|^2$$

Par l'inégalité de convexité, on sait que :

$$f(x_n) \leq f(x_*) + \langle \nabla f(x), x_n - x_* \rangle$$

Soit :

$$f(x_{n+1}) \leq f(x_*) + \langle \nabla f(x), x_n - x_* \rangle - \frac{t}{2}\|\nabla f(x)\|^2$$

$$f(x_{n+1}) \leq f(x_*) - \frac{1}{2t}(\|x_n - x_* - t\nabla f(x_n)\|^2 - \|x_n - x_*\|^2)$$

$$f(x_{n+1}) \leq f(x_*) + \frac{1}{2t}(-\|x_{n+1} - x_*\|^2 + \|x_n - x_*\|^2)$$

$$f(x_{n+1}) - f(x_*) \leq \frac{1}{2t}(\|x_n - x_*\|^2 - \|x_{n+1} - x_*\|^2)$$

En sommant sur les n allant de 0 à N-1 :

$$\sum_{k=0}^{N-1} f(x_{n+1}) - f(x_*) \leq \sum_{k=0}^{N-1} \frac{1}{2t}(\|x_n - x_*\|^2 - \|x_{n+1} - x_*\|^2)$$

On voit alors apparaître une somme télescopique dans le membre de droite :

$$\sum_{k=0}^{N-1} f(x_{n+1}) - f(x_*) \leq \frac{1}{2t}(\|x_0 - x_*\|^2 - \|x_n - x_*\|^2)$$

$$\frac{1}{n}\sum_{k=0}^{N-1} f(x_{n+1}) - f(x_*) \leq \frac{1}{2nt}(\|x_0 - x_*\|^2 - \|x_n - x_*\|^2)$$

$$f(x_{n+1}) - f(x_*) \leq \frac{1}{2nt}(\|x_0 - x_*\|^2 - \|x_n - x_*\|^2)$$

Finalement :

$$f(x_{n+1}) - f(x_*) \leq \frac{1}{2nt}(\|x_0 - x_*\|^2)$$

Ce qui conclut la démonstration. Pour plus d'informations, voir la référence [2].

2. **Second cas :**

Lemme 1

Soit f, une fonction continûment dérivable de paramètre de Lipschitz M et fortement convexe de paramètre m, alors

$$\forall x, y \quad (x-y)^T(\nabla f(x) - \nabla f(y)) \geq \frac{\alpha\beta\|x-y\|_2^2}{\beta+\alpha} + \frac{\|\nabla f(x) - \nabla f(y)\|_2^2}{\beta+\alpha}$$

Preuve, voir [1].

$$\begin{aligned}
\|x^{k+1} - x^*\|_2^2 &= \|x^k - x^* - t\nabla f(x^k)\|_2^2 && (1)\\
&= \|x^k - x^*\|_2^2 - 2t(x^k - x^*)^T\nabla f(x^k) + \|t\nabla f(x^k)\|_2^2 && (2)\\
&\leq \|x^k - x^*\|_2^2 + \|t\nabla f(x^k)\|_2^2 - \frac{2t}{m+M}(mM\|x^k - x^*\|_2^2 + \|\nabla f(x^k)\|_2^2) && (3)\\
& && (4)
\end{aligned}$$

d'après Lemme 1

$$\begin{aligned}
&= (1 - 2t\frac{mM}{m+M})\|x^k - x^*\|_2^2 + t(t - \frac{2}{m+M}\|\nabla f(x^k)\|_2^2) && (5)\\
&\leq (1 - 2t\frac{mM}{m+M})\|x^k - x^*\|_2^2 \quad \text{pour } 0 < t < \frac{2}{m+M} && (6)\\
&\leq (1 - 2t\frac{mM}{m+M})^{k+1}\|x^0 - x^*\|_2^2 && (7)
\end{aligned}$$

En utilisant le fait que la fonction f soit lipschitzienne, on obtient finalement :

$$f(x^k) - p^* \leq \frac{M}{2}(1 - 2t\frac{mM}{m+M})^k\|x^0 - x^*\|_2^2$$

On en conclut alors que f converge vers p^* en $O(c^k)$ avec $0 < c = 1 - 2t\frac{mM}{m+M} < 1$. La constante de convergence c est minimale pour $t = t_{opt} = \frac{2}{m+M}$. On obtient alors $c_{opt} = 1 - \frac{4mM}{(m+M)^2}$.
Pour plus d'informations, voir la référence [8].

Descente de gradient utilisant le backtracking :

$f : \Re^n \longrightarrow \Re$ est une fonction deux fois continûment dérivable et fortement convexe.
Donc $\exists\ 0 < m < M$ tels que

$$\forall x \in \Re^n \ mI \prec \nabla^2 f(x) \prec MI$$

D'une part, on a :

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{m}{2}\|y-x\|_2^2$$

On montre que le minimum du membre de droit est :

$$\tilde{y} = x - \frac{1}{m}\nabla f(x)$$

Donc :

$$f(\tilde{y}) \geq f(x) + \nabla f(x)^T(\tilde{y} - x) + \frac{m}{2}\|\tilde{y} - x\|_2^2 = f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2$$

Ce qui donne :

$$p^* \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2 \iff \|\nabla f(x)\|_2^2 \geq 2m(f(x) - p^*)$$

D'autre part on a :

$$f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{M}{2}\|y - x\|_2^2$$

Dans un premier temps, montrons que la condition de sortie de la boucle de backtracking :

$$\tilde{f}(t) = f(x - t\nabla f(x)) \leq f(x) - \alpha t\|\nabla f(x)\|_2^2$$

est satisfaite $\forall\ 0 \leq t \leq \frac{1}{M}$.
On note que

$$0 \leq t \leq \frac{1}{M} \Rightarrow -t + \frac{Mt^2}{2} \leq -\frac{t}{2}$$

En posant :

$$y = x - t\nabla f(x)$$

On a d'après (2) :

$$\tilde{f}(t) \leq f(x) - \nabla f(x)^T(t\nabla f(x)) + \frac{M}{2}\|y - x\|_2^2 \tag{8}$$

$$= f(x) + (-t + \frac{Mt^2}{2})\|\nabla f(x)\|_2^2 \tag{9}$$

$$\leq f(x) - \frac{t}{2}\|\nabla f(x)\|_2^2 \tag{10}$$

$$\leq f(x) - \alpha t\|\nabla f(x)\|_2^2 \text{ car } 0 < \alpha \leq \frac{1}{2} \tag{11}$$

Donc soit la condition est respectée pour $t = t_0 = 1$ soit elle l'est pour un $t = \beta t^-$ avec $t \leq \frac{1}{M} \leq t^-$ donc $t \geq \frac{\beta}{M}$
On a donc dans le premier cas :

$$f(x^+) \leq f(x) - \alpha\|\nabla f(x)\|_2^2$$

Et dans le second cas :

$$f(x^+) \leq f(x) - \frac{\alpha\beta}{M}\|\nabla f(x)\|_2^2$$

Donc, globalement :

$$f(x^+) \leq f(x) - \min(\alpha, \frac{\alpha\beta}{M})\|\nabla f(x)\|_2^2$$

$$\iff f(x^+) - p^* \leq f(x) - p^* - \min(\alpha, \frac{\alpha\beta}{M})\|\nabla f(x)\|_2^2$$

En utilisant (1), on obtient :

$$f(x^+) - p^* \leq (1 - \frac{2m\alpha}{M}\min(M, \beta))(f(x) - p^*)$$

Donc :

$$f(x^k) - p^* \leq c(f(x^k) - p^*) \tag{12}$$

$$\leq c^k(f(x^0) - p^*) \underset{k\to+\infty}{\longrightarrow} 0 \tag{13}$$

où $c = 1 - \frac{2m\alpha}{M}\min(M, \beta) < 1$

Pour plus d'informations, voir la référence [8].

Descente de gradient utilisant un pas optimal :

On reprend le début de la démonstration de la convergence du pas fixe avec

$t_k = \underset{t}{\operatorname{argmin}}\ f(x_{k-1} - t\nabla f(x_{k-1})$.

$$\|x^{k+1} - x^*\|_2^2 \leq (1 - 2t_k\frac{mM}{m+M})\|x^k - x^*\|_2^2 \quad \text{pour } 0 < t_k < \frac{2}{m+M} \tag{14}$$

$$\leq \prod_{i=0}^{k}(1 - 2t_k\frac{mM}{m+M})\|x^0 - x^*\|_2^2 \tag{15}$$

D'où :

$$f(x^k) - p^* \leq \frac{M}{2}\|x^0 - x^*\|_2^2 \prod_{i=0}^{k}(1 - 2t_k\frac{mM}{m+M})$$

Tentons d'approcher $t_k = \underset{t}{\operatorname{argmin}}\ f(x_{k-1} - t\nabla f(x_{k-1}$.

$$\partial_t f(x - t\nabla f(x)) = -\nabla f(x)^T\nabla f(x - t\nabla f(x)) \tag{16}$$

$$\underset{\nabla f(x)\longrightarrow 0}{=} -\nabla f(x)^T(\nabla f(x) - t\nabla^2 f(x)) \tag{17}$$

$$= -\|\nabla f(x)\|_2^2 + t\|\nabla^2 f(x)^{1/2}\nabla f(x)\|_2^2 \tag{18}$$

$$\partial_t f(x - t\nabla f(x)) = 0 \Leftrightarrow t = \frac{\|\nabla f(x)\|_2^2}{\|\nabla^2 f(x)^{1/2}\nabla f(x)\|_2^2}$$

Finalement, on obtient un encadrement de t_k $\forall\ k : \frac{1}{M} < t_k < \frac{1}{m}$. Finalement obtient

$$f(x^k) - p^* \leq \frac{M}{2}(1 - 2\frac{m}{m+M})^k \|x^0 - x^*\|_2^2$$

Donc :

$$f(x^k) - p^* \leq \frac{M}{2} c^k \|x^0 - x^*\|_2^2$$

où $c = 1 - 2\frac{m}{m+M}$

Pour plus d'informations, voir la référence [8].

1.6 Illustration dans la cadre de la minimisation de fonctions quadratiques

Testons les algorithmes proposés ci-dessus pour minimiser la fonction quadratique $f : \Re^n \longrightarrow \Re$ suivante :

$$f(x) = x^T Ax + b^T x + c$$

où $A \in \Re^{n \times n}$ est symétrique, $b \in \Re^n$ et $c \in \Re$ sont générés aléatoirement.

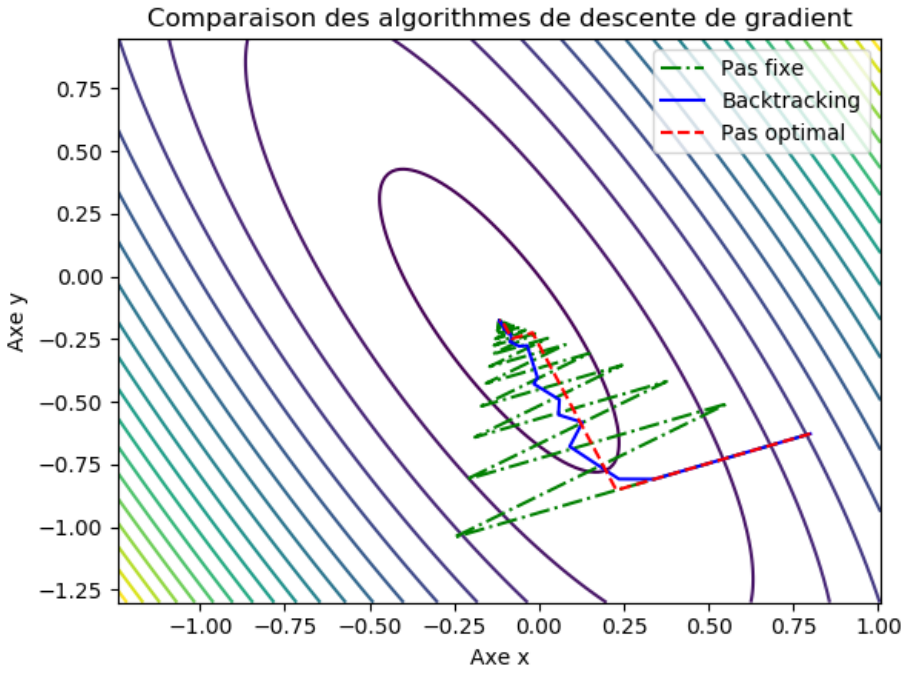

FIGURE 1 – Observation des vitesses de convergences dans le cas d'un problème en 2 dimensions

En grande dimension, $n = 100$. Si on affiche sur une échelle logarithmique l'écart en norme entre l'estimation à l'itération k et la solution analytique, on remarque qu'après un état de transition, la convergence vers la solution pour chacun des algorithmes se fait linéairement. Cela signifie que conformément à l'étude théorique, la convergence se fait bien en $O(c^k)$. En calculant les constantes de convergences expérimentales (qui correspond alors à dix à la puissance la pente observée sur le graphique) et la limite théorique, on obtient les résultats suivants :

Algorithme	Constante de convergence expérimentale	Limite supérieure de convergence théorique
Gradient à pas fixe	0.998517	0.999401
Gradient backtracking	0.999661	0.999925
Gradient à pas optimal	0.998918	0.999700

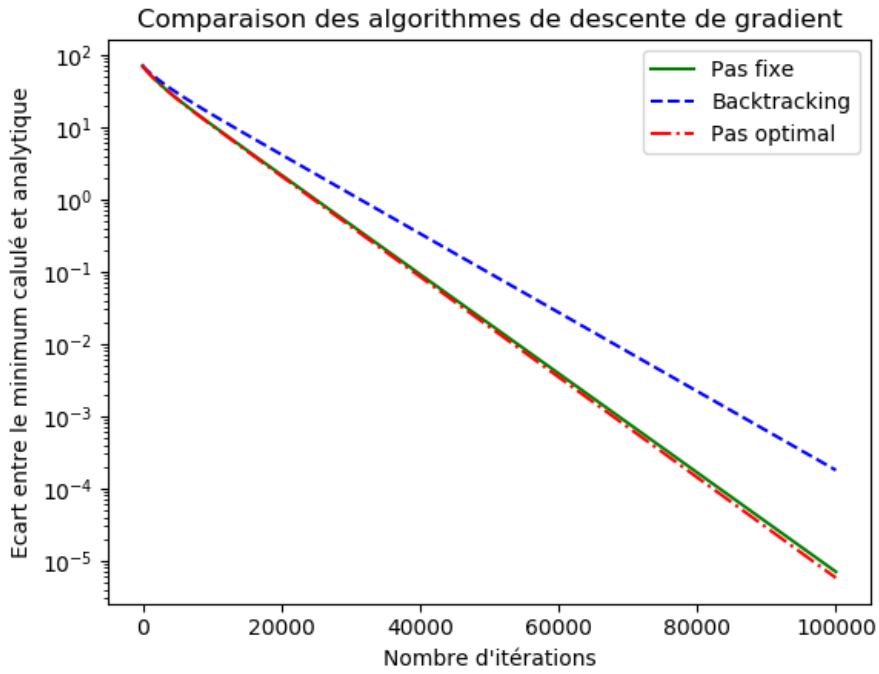

FIGURE 2 – Comparaison des vitesses de convergences à l'aide des constantes de convergence

Algorithme	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-5}$
Gradient Pas Fixe	2.307 s 43 058 itér.	6.399 s 111 070 itér.	22.44 s 379 836 itér.
Gradient Pas Optimal	4.016 s 41 989 itér.	10.34 s 109 259 itér.	35.92 s 371 996 itér.
Gradient Backtracking	5.650 s 56 227 itér.	14.23 s 145 185 itér.	51.16 s 504 114 itér.

En comparant ces deux valeurs pour chaque algorithme, on remarque, conformément avec ce que prédit la théorie, que les constantes expérimentales sont systématiquement plus faibles que la limite théorique.

De plus, on remarque que la constante de convergence de l'algorithme à pas optimal est légèrement inférieure à celle de l'algorithme de gradient à pas fixe optimal. Par conséquent, il vaut mieux implémenter l'algorithme à pas fixe optimal plutôt que l'algorithme à pas optimal car ce

dernier nécessite en plus un coût de calcul plus élevé en raison du problème d'optimisation qu'il doit résoudre à chaque itération. Quant à l'algorithme du gradient backtracking, celui-ci ne doit être implémenté en raison de sa lenteur que si nous n'avons aucune connaissance sur la constance de Lipschitz, le paramètre de forte convexité de la fonction et que nous ne sommes pas capable de résoudre explicitement le problème d'optimisation de l'algorithme du gradient optimal. Ainsi, les conclusions de l'étude expérimentale rejoignent celles de l'étude théorique.

2 Méthodes d'optimisation de fonctions non dérivables

2.1 Notions et résulats utilisés

1. Classe de fonctions propres fermés convexes (CCP) :
   CCP est l'acronyme de Convex Closed proper, c'est-à-dire convexe fermé et propre. Une fonction est convexe si et seulement si en tout point x de son domaine de définition, on a l'inégalité suivante qui est vérifiée :

   $$\forall\ x, y \in \mathrm{dom} f \text{ et } \theta \in [0,1]\ f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$$

   Une fonction f est fermée si et seulement si l'une des propriétés suivantes est vérifiée :
   - Son épigraphe
     $$\mathrm{epi} f = \{(x,t) \in \Re^{d+1} | f(x) \leq t\}$$
     est fermé.
   - f est une fonction semi-continue inférieure c'est-à-dire que $\forall\ x_0 \in \mathrm{dom} f$ et $\forall\ \varepsilon > 0$, il existe un voisinage U de x_0 tel que
     $$\forall x \in U, f(x) \geq f(x_0) - \varepsilon$$

   Une fonction est propre si son ensemble de définition est non vide.
2. La sous-différentielle :
   La sous-différentielle $\partial f(x)$ d'une fonction f en x est l'ensemble :
   $$\partial f(x) = \{y | \forall z \in \Re^d f(z) \geq f(x) + y^T (z - x)\}$$
3. Fonctions non expansives et contractantes :
   Une fonction non expansive est lipschitzienne de paramètre $L = 1$. Une fonction contractante est lipschitzienne de paramètre $0 < L < 1$.
4. Opérateur moyenné :
   Un opérateur F est moyenné si et seulement s'il peut s'écrire sous la forme $F = (1-\theta)I + \theta G$ avec $\theta \in [0, 1[$ et G une fonction non expansive.
5. Opérateur proximal :
   $$\mathrm{prox}_{\lambda, f}(\nu) = \underset{x}{\mathrm{argmin}}\ (f(x) + \frac{1}{2\lambda}\|x - \nu\|_2^2)$$

2.2 Hypothèses et résultats

Pour chacun des algorithmes suivants, on choisit un critère d'arrêt à tester à chaque itération.

Gradient proximal :

Cet algorithme permet de trouver le minimum de la somme des deux fonctions ayant le même domaine de définition, f est dérivable et g ne l'est pas. Pour cela, l'algorithme évalue à chaque itération l'opérateur proximal de la fonction non dérivable g au point de l'itération précédente diminué de l'opposé du gradient de la fonction dérivable f.

Hypothèses :
f et g sont des fonctions CCP de $\Re^d$ dans $\Re$. f est continûment dérivable de constante de lipschitz $0 < M$. Le pas $t \in]0, \frac{1}{M}]$.

Résultat :
L'algorithme converge vers p^* en $O(\frac{1}{n})$.

Gradient proximal accéléré :

Il s'agit du même algorithme que précédemment avec une étape supplémentaire à chaque itération permettant d'accroître la vitesse de convergence de l'algorithme.

Hypothèses :
f et g sont des fonctions CCP de $\Re^d$ dans $\Re$. f est continûment dérivable de constante de lipschitz $0 < M$. Le pas $t \in]0, \frac{1}{M}]$.

Résultat :
L'algorithme converge vers p^* en $O(\frac{1}{n^2})$.

Alternating Direction Method of Multipliers (ADMM) :

Cet algorithme permet de minimiser la somme des deux fonctions f et g non dérivables ayant des domaines de défintion potentiellement différents et sous la contrainte que les arguments des deux fonctions soient reliés par une relation linéaire. L'algorithme consiste alors à calculer successivement à chaque itération le minimum du lagrangien augmenté des deux fonctions f et g.

Hypothèses :
f et g sont des fonctions CCP de $\Re^d$ dans $\Re$.

Résultat :
L'algorithme converge vers p^* en $O(\frac{1}{n})$.

2.3 Description des algorithmes

Gradient proximal :

On cherche à minimiser $f(x) + g(x)$.

Algorithm 4 Gradient proximal

repeat
$x \leftarrow \text{prox}_{\lambda g}(x - \lambda \nabla f(x))$
until critère d'arrêt

Gradient proximal accéléré :

On cherche à minimiser $f(x) + g(x)$.

Algorithm 5 Gradient proximal accéléré

repeat
$x^k \leftarrow \text{prox}_{\lambda g}(y^k - \lambda \nabla f(y^k))$
$y^k \leftarrow x^k + \frac{k-1}{k+2}(x^k - x^{k-1})$
until critère d'arrêt

Alternating Direction Method of Multipliers (ADMM) :

On cherche à minimiser $f(x) + g(z)$ sous la contrainte $Ax + Bz = c$.
On pose $L_\rho(x, y, z) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2$, le lagrangien augmenté du problème d'optimisation avec $\rho > 0$.

Algorithm 6 Alternating Direction Method of Multipliers (ADMM)

repeat
$x^{k+1} \leftarrow \underset{x}{\text{argmin}}\ L_\rho(x, z^k, y^k)$
$z^{k+1} \leftarrow \underset{x}{\text{argmin}}\ L_\rho(x^{k+1}, z, y^k)$
$y^{k+1} \leftarrow y^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$
until critère d'arrêt

2.4 Preuves de la convergence des algorithmes

Gradient proximal :

$$x^* = \underset{x}{\operatorname{argmin}}\ f(x) + g(x) \Leftrightarrow 0 \in \nabla f(x^*) + \partial g(x^*)$$

L'algorithme du gradient proximal consiste à appliquer l'algorithme foward backward à $\nabla f + \partial g$. D'après [4], $\nabla f + \partial g$ sont alors deux fonctions multivaluées monotones maximales car f et g sont CCP.

Soit A et B deux fonctions monotones maximale. Supposons A univaluée et $\alpha > 0$.

$$0 \in (A+B)(x) \Leftrightarrow 0 \in (I+\alpha B)(x) - (I-\alpha A)(x) \tag{19}$$
$$\Leftrightarrow (I-\alpha A)(x) \in (I+\alpha B)(x) \tag{20}$$
$$\Leftrightarrow x \in (I+\alpha B)^{-1}(I-\alpha A)(x) \tag{21}$$

Toujours d'après [4], on montre que $(I+\alpha\partial g)^{-1}(I-\alpha\nabla f)$ est un opérateur moyenné en montrant que les opérateurs forward et backward sont des opérateurs moyennés et que leur composition l'est aussi. Ainsi on prouve la convergence des itérations de points fixes de l'algoritme forward backward vers un point fixe s'il existe.

Montrons que $\text{prox}_{\lambda,g} = (I + t\partial g)^{-1}$:

$$\text{Soit } z = \text{prox}_{t,g}(x) \Leftrightarrow z = \underset{u \in domg}{\operatorname{argmin}}\ (g(u) + \frac{1}{2t}\|u-x\|_2^2) \tag{22}$$
$$\Leftrightarrow 0 \in \partial_z(g(z) + \frac{1}{2t}\|z-x\|_2^2) \tag{23}$$
$$\Leftrightarrow 0 \in \partial g(z) + \frac{1}{t}(z-x) \tag{24}$$
$$\Leftrightarrow z \in (I+t\partial g)^{-1}(x) \tag{25}$$

Finalement on retrouve bien l'opération qu'effectue l'algorithme du gradient proximal à chaque itération c'est-à-dire, $x^+ = \text{prox}_{\lambda,g}(x - \nabla f(x))$.

Analyse de la convergence :

f est continûment dérivable de constante de Lipschitz L. On a donc la propriété :

$$\forall x, y \in \text{dom} f\ \ f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}\|y-x\|_2^2$$

En posant $y = x - tG_t(x)$, où $G_t(x) = \frac{1}{t}(x - \text{prox}_{t,g}(x - t\nabla f(x)))$ on obtient :

$$f(x - tG_t(x)) \leq f(x) - t\nabla f(x)^T(G_t(x)) + \frac{Lt^2}{2}\|G_t(x)\|_2^2$$

avec $G_t(x) = \frac{1}{t}(x - \text{prox}_{t,g}(x - t\nabla f(x)))$ On prend alors $0 < t \leq \frac{1}{L}$, ainsi on a :

$$f(x - tG_t(x)) \leq f(x) - t\nabla f(x)^T(G_t(x)) + \frac{t}{2}\|G_t(x)\|_2^2$$

En utilisant la convexité de g, il vient :

$$\forall z \; g(x - tG_t(x)) \leq g(z) + \nu^T(x - z - tG_t(x))$$

avec $\nu \in \partial g(x - tG_t(x))$.
On sait que $u = \text{prox}_{t,g}(v) \Leftrightarrow v - u \in \partial g(u)$ donc avec $u = x - tG_t(x)$ et $v = x - t\nabla f(x)$, $t(G_t(x) - \nabla f(x)) \in \partial g(x - tG_t(x))$ on prend donc $\nu = G_t(x) - \nabla f(x)$.

En sommant les deux dernières inégalités, on obtient :

$$(f + g)(x - tG_t(x)) \leq f(z) + g(z) + G_t(x)^T(x - z) - \frac{t}{2}\|G_t(x)\|_2^2$$

En posant $z = x$, $x = x^k$ et $x - tG_t(x) = x^{k+1}$ on a :

$$(f + g)(x^{k+1}) \leq (f + g)(x^k) - \frac{t}{2}\|G_t(x)\|_2^2$$

Donc $(f + g)(x^k)$ est une suite décroissante.

En posant $z = x^*$, $x = x^k$ et $x - tG_t(x) = x^{k+1}$ on a :

$$\begin{aligned}(f + g)(x^{k+1}) - (f + g)^* &\leq G_t(x)^T(x^k - x^*) - \frac{t}{2}\|G_t(x^k)\|_2^2 && (26)\\ &= \frac{1}{2t}(\|x^k - x^*\|_2^2 - \|x - x^* - tG_t(x^k)\|_2^2) && (27)\\ &= \frac{1}{2t}(\|x^k - x^*\|_2^2 - \|x^{k+1} - x^*\|_2^2) && (28)\end{aligned}$$

En sommant les inégalités pour i allant de 0 à k :

$$\sum_{i=0}^{k}(f + g)(x_{i+1}) - (f + g)(x_*) \leq \sum_{i=0}^{k}\frac{1}{2t}(\|x_i - x_*\|^2 - \|x_{i+1} - x_*\|^2)$$

On voit alors apparaître une somme téléscopique dans le membre de droite :

$$\sum_{i=0}^{k}(f + g)(x_{i+1}) - (f + g)(x_*) \leq \frac{1}{2t}(\|x_0 - x_*\|_2^2 - \|x_{k+1} - x_*\|^2) \leq \frac{1}{2t}\|x_0 - x_*\|_2^2$$

Comme $(f + g)(x^k)$ est décroissante, on a finalement :

$$(f + g)(x_{k+1}) - (f + g)(x_*) \leq \frac{1}{2t(k + 1)}\|x_0 - x_*\|_2^2$$

Pour plus d'informations, voir [9].

Gradient proximal accéléré :

Analyse de la convergence :

f est continûment dérivable de constante de Lipschitz L. On a donc la propriété :

$$\forall x, y \in \mathrm{dom} f \ f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}\|y - x\|_2^2$$

En posant $y = x - tG_t(x)$, où $G_t(x) = \frac{1}{t}(x - \mathrm{prox}_{t,g}(x - t\nabla g(x)))$ on obtient :

$$f(x - tG_t(x)) \leq f(x) - t\nabla f(x)^T (G_t(x)) + \frac{Lt^2}{2}\|G_t(x)\|_2^2$$

avec $G_t(x) = \frac{1}{t}(x - \mathrm{prox}_{t,g}(x - t\nabla f(x)))$ On prend alors $0 < t \leq \frac{1}{L}$, ainsi on a :

$$f(x - tG_t(x)) \leq f(x) - t\nabla f(x)^T (G_t(x)) + \frac{t}{2}\|G_t(x)\|_2^2$$

Considérons les notations suivantes :

$$\theta_k = \frac{2}{1+k}$$

$$\nu^k = x^{k-1} + \frac{1}{\theta_k}(x^k - x^{k-1}) \ \text{ avec } \nu^0 = x^0$$

On obtient alors les relations suivantes :

$$y^k = (1 - \theta_{k+1})x^k + \theta_{k+1}\nu^k$$

$$\nu^k = x^{k-1} + \frac{1}{\theta_k}(y^{k-1} - tG_t(y^{k+1} - x^{k-1}) = \nu^{k-1} - \frac{t}{\theta_k}G_t(y^{k-1} - x^{k-1})$$

$$\frac{1 - \theta_k}{\theta_k^2} \leq \frac{1}{\theta_{k-1}^2}$$

Pour alléger les notations, posons maintenant :

$$x = x^{k-1} \ , \ x^+ = x^k \ , \ y = y^{k-1} \ , \ \nu = \nu^{k-1} \ , \ \nu^+ = \nu^k \ , \ \theta = \theta_k$$

On choisit t tel que pour tout z, l'inégalité suivante est vérifiée :

$$(f+g)(x^+) = (f+g)(y - tG_t(y)) \leq (f+g)(z) + G_t(y - z) - \frac{t}{2}\|G_t(y)\|_2^2$$

On choisit alors $z = (1-\theta)x + \theta x^*$ et d'après l'inégalité de convexité,

$$f(x^+) \leq (1-\theta)(f+g)(x) + \theta f^* + G_t(y)^T(y - (1-\theta)x - \theta x^*) - \frac{t}{2}\|G_t(y)\|_2^2 \tag{29}$$

$$= (1-\theta)(f+g)(x) + \theta f^* + \frac{\theta^2}{2t}(\frac{2t}{\theta}G_t(y)^T(\nu - x^*) - \frac{t^2}{\theta^2}\|G_t(y)\|_2^2) \tag{30}$$

$$= (1-\theta)(f+g)(x) + \theta f^* + \frac{\theta^2}{2t}(\frac{2t}{\theta}\|\nu - x^*\| - \|\nu - x^* + \frac{t^2}{\theta}G_t(y)\|_2^2) \tag{31}$$

$$= (1-\theta)(f+g)(x) + \theta f^* + \frac{\theta^2}{2t}(\frac{2t}{\theta}\|\nu - x^*\| - \|\nu^+ - x^*\|_2^2) \qquad (*) \tag{32}$$

$$(*) \iff \frac{1}{\theta^2}(f(x^+) - f^*) \leq \frac{1-\theta}{\theta^2}(f(x) - f^*) + \frac{1}{2t}(\|\nu - x^*\|_2^2 - \|\nu^+ - x^*\|_2^2) \tag{33}$$

$$\iff \frac{1}{\theta_k^2}(f(x^k) - f^*) + \|\nu^k - x^*\|_2^2 \leq \frac{1-\theta_k}{\theta_k^2}(f(x^{k-1}) - f^*) + \frac{1}{2t}\|\nu^{k-1} - x^*\|_2^2 \tag{34}$$

En sommant les inégalités de 0 à k, on obtient alors :

$$(*) \iff \frac{1}{\theta_k^2}(f(x^k) - f^*) + \|\nu^k - x^*\|_2^2 \leq \frac{1-\theta_1}{\theta_1^2}(f(x^0) - f^*) + \frac{1}{2t}\|\nu^0 - x^*\|_2^2 \tag{35}$$

$$\iff \frac{1}{\theta_k^2}(f(x^k) - f^*) \leq \frac{1-\theta_1}{\theta_1^2}(f(x^0) - f^*) + \frac{1}{2t}\|\nu^0 - x^*\|_2^2 = \frac{1}{2t}\|\nu^0 - x^*\|_2^2 \tag{36}$$

Finalement on a :

$$f(x_k) - f^* \leq \frac{2}{t(k+1)^2}\|x^0 - x^*\|_2^2$$

Pour plus d'informations, voir [9].

Alternating Direction Method of Multipliers (ADMM) :

Descente de gradient pour optimisation sous contraintes

$$\text{minimiser : } f(x)$$

$$\text{sous contrainte : } Ax = b$$

Le lagrangien s'écrit : $L(x, y) = f(x) + y^T(Ax - b)$

La fonction duale s'écrit : $g(y) = \inf_x L(x, y)$

Le problème dual est : $\max g(y)$

Dans le cas où f est une fonction fortement convexe, la solution du dual est la même que la solution du primal. On peut alors appliquer la méthode de descente de gradient au dual.

On a alors une itération de la méthode du gradient qui est :

$$y^+ = y + t\nabla g(y)$$

Avec $\nabla g(y) = A\tilde{x} - b$ et $\tilde{x} = \underset{x}{\text{argmin}} \ \ L(x, y)$

L'algorithme de descente de gradient est donc le suivant :

$$x^{k+1} = \underset{x}{\text{argmin}}\ L(x, y^k)$$

$$y^{k+1} = y^k + t^k(Ax^{k+1} - b)$$

La méthode des multiplieurs

Le lagrangien augmenté permet de donner de la robustesse à la méthode de descente de gradient décrite précédemment afin d'avoir la convergence sans que la fonction f ne soit fortement convexe. On définit le lagrangien augmenté de la manière suivante :

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + \frac{\rho}{2}\|Ax - b\| \ , \ \rho > 0$$

On remarque en prenant $\rho = 0$ que l'on retrouve le Lagrangien usuel. Le Lagrangien augmenté peut être considéré comme le Lagrangien (non augmenté) du problème suivant qui est exactement équivalent au problème de départ.

$$\text{minimiser} : f(x) + \frac{\rho}{2}\|Ax - b\|$$

$$\text{sous contrainte} : Ax = b$$

La méthode des multiplieurs revient à appliquer la méthode de descente de gradient en prenant le lagrangien augmenté, et un pas $t = \rho$. On obtient alors :

$$x^{k+1} = \underset{x}{\text{argmin}} \ L_\rho(x, y^k)$$

$$y^{k+1} = y^k + \rho(Ax^{k+1} - b)$$

Montrons que la méthode des multiplieurs converges :
Les conditions d'optimalités sont :

$$Ax^* - b = 0$$

$$\nabla f(x^*) + A^T(y^k + \rho(Ax^{k+1} - b)) = 0$$

Par définition, x^{k+1} minimise $L_\rho(x^{k+1}, y^k)$.

$$\begin{aligned} 0 &= \nabla_x L_\rho(x^{k+1}, y^k) && (37)\\ &= \nabla f(x^{k+1}) + A^T(y^k + \rho(Ax^{k+1} - b)) && (38)\\ &= \nabla f(x^{k+1}) + A^T y^{k+1} && (39) \end{aligned}$$

Donc à chaque itération, (x^{k+1}, y^{k+1}) est admissible pour le problème dual. De plus, comme $y^{k+1} - y^k \underset{k\to+\infty}{\longrightarrow} 0$, alors $Ax^{k+1} - b \underset{k\to+\infty}{\longrightarrow} 0$. Donc les conditions d'optimalités sont asymptotiquement respectées donc $x^k \underset{k\to+\infty}{\longrightarrow} x^*$.

Alternating Direction Method of Multipliers

$$\text{minimiser} : f(x) + g(z)$$

$$\text{sous contrainte} : Ax + Bz = c$$

Les itérations de l'ADMM sont :

$$x^{k+1} = \underset{x}{\text{argmin}} \ L_\rho(x, z^k, y^k)$$

$$z^{k+1} = \underset{z}{\operatorname{argmin}}\ L_\rho(x^{k+1}, z, y^k)$$

$$y^{k+1} = y^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$$

La méthode ADMM est alors très similaire à la méthode des multiplieurs à ceci près que dans la méthode des multiplieurs, la minimisation de $L_\rho(x, z, y)$ par rapport à x et z se fait en même temps.

Les conditions d'optimalités sont les suivantes :

$$Ax^* + Bz^* = c$$

$$0 \in \partial f(x^*) + A^T y^*$$

$$0 \in \partial g(z^*) + B^T y^*$$

Puisque z^{k+1} minimize $L_\rho(x^{k+1}, z, y^k)$, on a alors :

$$0 \in \partial g(z^{k+1}) + B^T y^k + \rho B^T(Ax^{k+1} + Bz^{k+1} - c) \tag{40}$$

$$\in \partial g(z^{k+1}) + B^T y^{k+1} \tag{41}$$

Donc z^{k+1} et y^{k+1} vérifient systématiquement la condition 3 d'optimalité.

Puisque x^{k+1} minimize $L_\rho(x, z^k, y^k)$, on a alors :

$$0 \in \partial f(x^{k+1}) + A^T y^k + \rho A^T(Ax^{k+1} + Bz^{k+1} - c) \tag{42}$$

$$\in \partial f(x^{k+1}) + A^T(y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) + \rho B(z^k - z^{k+1})) \tag{43}$$

$$\in f(x^{k+1}) + A^T y^{k+1} + \rho A^T B(z^k - z^{k+1}) \tag{44}$$

Finalement les conditions d'optimalités sont respectées si les quantités $s^{k+1} = \rho A^T B(z^k - z^{k+1})$ et $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$ tendent vers 0 [5].

Lien avec la théorie des opérateurs monotones

On peut réécrire le lagrangien augmenté de la façon suivante :

$$L_\rho(x, z, y) = f(x) + g(z) + \frac{\rho}{2}\|Ax + Bz - c + \frac{1}{\rho}y\|_2^2 - \|\frac{1}{\rho}y\|_2^2$$

En posant $u = \frac{1}{\rho}y$, on obtient alors :

$$L_\rho(x, z, u) = f(x) + g(z) + \frac{\rho}{2}\|Ax + Bz - c + u\|_2^2 - \|u\|_2^2$$

Lorsque $A = I$,

$$x^{k+1} = \underset{x}{\operatorname{argmin}}\ L_\rho(x, z^k, u^k) = \underset{x}{\operatorname{argmin}}\ (f(x) + \frac{\rho}{2}\|x + Bz^k - c + u^k\|_2^2) = \operatorname{prox}_{\rho,f}(-Bz^k + c - u^k)$$

De même lorsque $B = I$,

$$z^{k+1} = \underset{z}{\operatorname{argmin}}\ L_\rho(x^{k+1}, z, u^k) = \underset{z}{\operatorname{argmin}}\ (g(z) + \frac{\rho}{2}\|x^{k+1} + z - c + u^k\|_2^2) = \operatorname{prox}_{\rho,g}(-x^{k+1} + c - u^k)$$

2.5 Illustration dans la cadre de la minimisation de fonctions de type LASSO

Testons les algorithmes proposés ci-dessus pour minimiser un problème de type LASSO :

$$f(x) = \|Ax - b\|_2^2 + \|x\|_1$$

où $A \in \Re^{n \times n}$ est symétrique et $b \in \Re^n$ sont générées aléatoirement. Posons également $0 < M$ la constante de Lipschitz associée à la fonction $x \longmapsto \|Ax - b\|_2^2$.

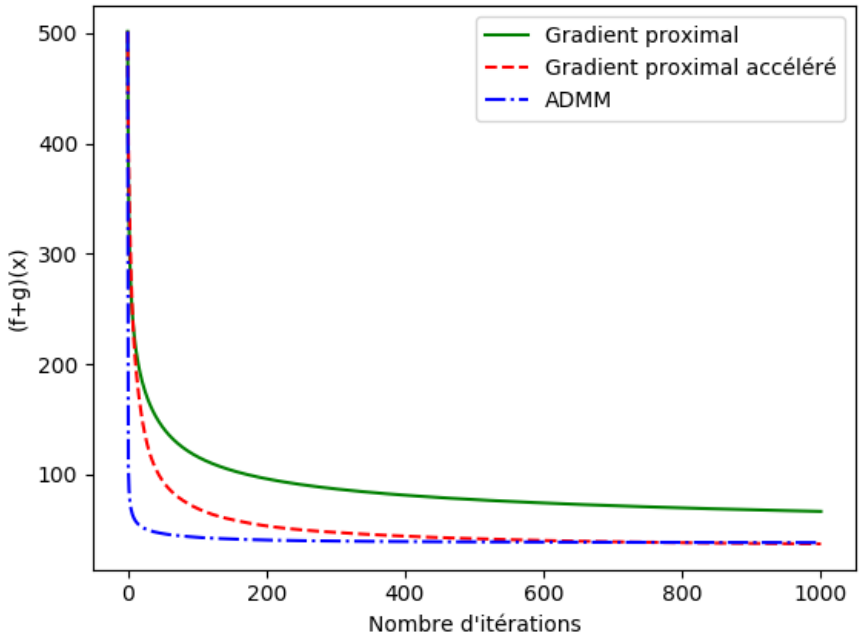

FIGURE 3 – Comparaison des vitesses de convergence pour la résolution d'un problème de type LASSO pour un problème de taille $n = 200$

L'étude théorique de la convergence des trois algorithmes implémentés a permis de mettre en évidence une borne supérieure dans la convergence de $f(x^k) - p^*$ vers 0. Ainsi, pour rappel, l'algorithme du gradient proximal et l'ADMM convergent en $O(1/k)$ et l'algorithme du gradient proximal accéléré converge $O(1/k^2)$. Essayons alors de déterminer les vitesses de convergences expérimentales de ces algorithmes dans le cadre de la minimisation d'un problème de type LASSO. On fait l'hypothèse en cohérence avec l'étude théorique précédente avec les algorithmes du gradient proximal, une suite $(u^k)_{k \geq 0}$ proportionnelle à $f(x^k) - p^*$ converge vers 0 en $O(1/k^n)$. Nous allons donc mettre en évidence que cette hypothèse est tout à fait crédible et nous allons déterminer la valeur de l'exposant n.

On a donc fait l'hypothèse de l'égalité suivante :

$$u^k \underset{k>>1}{=} \frac{1}{k^n} \iff \log\left(\frac{1}{u^k}\right) \underset{k>>1}{=} n \ \log(k)$$

Ainsi $\log\left(\frac{1}{u^k}\right)$ croît linéairement avec $\log(k)$. La valeur de l'exposant recherché est donné par la pente de cette droite.

Algorithme de descente de gradient proximale :

En appliquant l'algorithme du gradient proximal à un problème de type LASSO, on effectue l'itération suivante :

$$x^{k+1} = \text{prox}_{t,\|.\|_1}(x^k - 2tA^T(Ax^k - b))$$

D'après l'étude théorique,

$$f(x_k) - f^* \leq \frac{1}{2tk}\|x_0 - x_*\|_2^2 \iff \log\left(\frac{\|x_0 - x_*\|_2^2}{2(f(x_k) - f^*)}\right) \geq \log(k)$$

En traçant alors $\log\left(\frac{\|x_0 - x_*\|_2^2}{2t(f(x_k) - f^*)}\right)$ avec le pas maximal $t = \frac{1}{M}$, en fonction de k et avec une échelle des abscisses logarithmiques, on obtient la figure suivante :

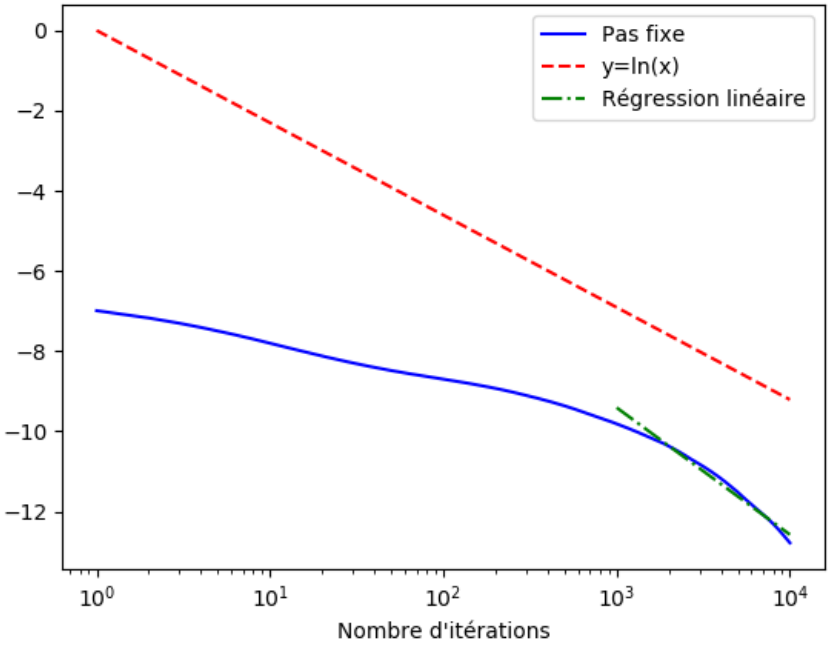

FIGURE 4 – Comparaison entre $\log\left(\frac{\|x_0 - x_*\|_2^2}{2t(f(x_k) - f^*)}\right)$ en vert clair et $\log(k)$ en vert foncé en fonction des itérations

Conformément à ce que prédit la théorie, $u_k = \log\left(\frac{\|x_0 - x_*\|_2^2}{2t(f(x_k)-f^*)}\right) \geq \log(k)$. De plus, après un régime transitoire d'environ un millier d'itérations, la suite u_k tend vers une suite affine dont la pente nous donne la valeur de l'exposant recherché. Ainsi, dans le cas de l'algorithme de descente de gradient proximale, une régression linéaire donne une valeur de pente de 1.3. Ainsi, on peut en conclure que l'algorithme de descente de gradient proximale converge vers la solution du problème avec une vitesse en $O(\frac{1}{k^{1,3}})$. La qualité de l'estimation est confirmée par le coefficient de détermination de la la régression $r = 0.98$, proche de 1.

Algorithme de descente de gradient proximale accélérée :

En appliquant l'algorithme du gradient proximal accéléré à un problème de type LASSO, on effectue les itérations suivante :

$$y^{k+1} = x^k + \frac{k-1}{k+2}(x^k - x^{k-1})$$

$$x^{k+1} = \text{prox}_{t,\|\cdot\|_1}(y^{k+1} - 2tA^T(Ay^{k+1} - b))$$

D'après l'étude théorique,

$$f(x_k) - f^* \leq \frac{2}{tk}\|x_0 - x_*\|_2^2 \Longleftrightarrow \log\left(\frac{2\|x_0 - x_*\|_2^2}{2(f(x_k) - f^*)}\right) \leq \log(k)$$

En traçant alors $\log\left(\frac{\|x_0 - x_*\|_2^2}{(f(x_k)-f^*)}\right)$ avec le pas maximal $t = \frac{1}{M}$ en fonction de k et avec une échelle des abscisses logarithmiques, on obtient la figure suivante :

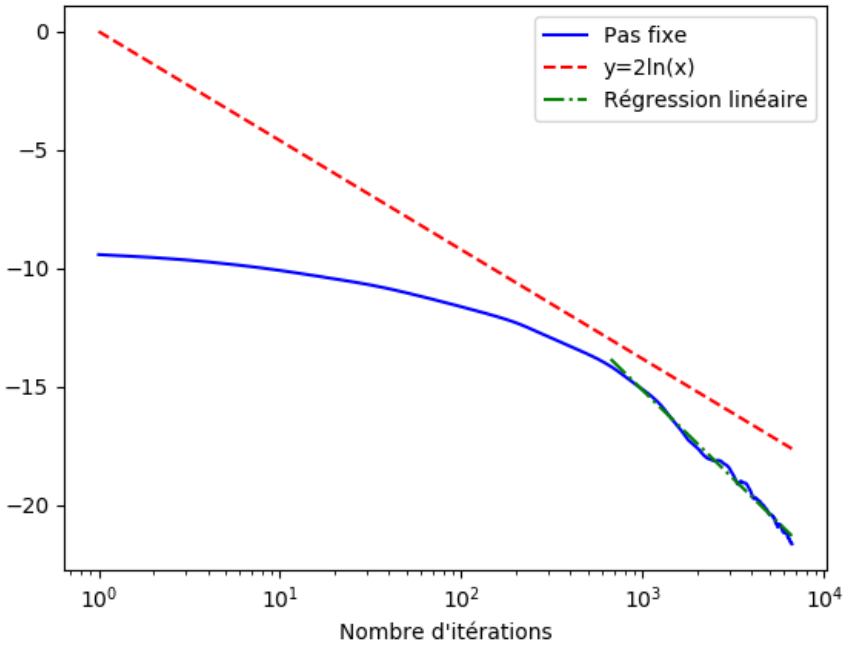

FIGURE 5 – Comparaison entre $\log\left(\frac{2\|x_0-x_*\|_2^2}{t(f(x_k)-f^*)}\right)$ en vert clair et $2\log(k)$ en vert foncé en fonction des itérations

Une fois encore et conformément à ce que prédit la théorie, $u_k = \log\left(2\frac{\|x_0-x_*\|_2^2}{t(f(x_k)-f^*)}\right) \geq 2\log(k)$. De plus, après un régime transitoire de quelques centaines d'itérations, la suite u_k tend vers une suite affine dont la pente nous donne la valeur de l'exposant recherché. Ainsi, dans le cas de l'algorithme de descente de gradient proximale, une régression linéaire donne une valeur de pente de 4.3. Ainsi, on peut en conclure que l'algorithme de descente de gradient proximale converge vers la solution du problème avec une vitesse en $O(\frac{1}{k^{4,3}})$. La qualité de l'estimation est confirmée par le coefficient de détermination de la régression $r = 0.98$, proche de 1.

ADMM :

En appliquant l'algorithme ADMM à un problème de type LASSO, on cherche à résoudre le problème suivant :

$$\min \|Ax - b\|_2^2 + \|z\|_1$$
$$\text{s.c } x - z = 0$$

Pour cela, on effectue les itérations suivante :

$$x^{k+1} = (A^TA + \rho I)^{-1}(A^T + \rho z^k - y^k)$$

$$z^{k+1} = \text{prox}_{t,\|.\|_1}(x^{k+1} + \frac{y^{k+1}}{\rho})$$

$$y^{k+1} = y^k + \rho(x^{k+1} - z^{k+1})$$

En traçant alors $\log\left(\frac{\|x_0 - x_*\|_2^2}{(f(x_k) - f^*)}\right)$ avec une échelle des abscisses logarithmiques, on obtient la figure suivante :

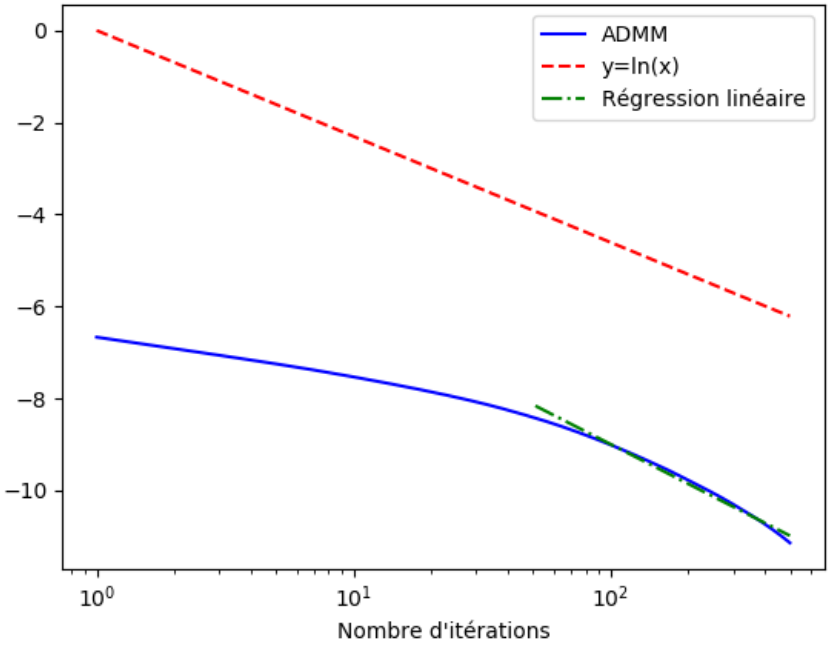

FIGURE 6 – Comparaison entre $\log\left(\frac{\|x_0 - x_*\|_2^2}{(f(x_k) - f^*)}\right)$ en vert clair et $\log(k)$ en vert foncé en fonction des itérations

Une fois encore et conformément à ce que prédit la théorie, $u_k = \log\left(\frac{\|x_0 - x_*\|_2^2}{(f(x_k) - f^*)}\right) \geq \log(k)$. De plus, après un régime transitoire de quelques centaines d'itérations, la suite u_k tend vers une suite affine dont la pente nous donne la valeur de l'exposant recherché. Ainsi, dans le cas de l'algorithme ADMM, une régression linéaire donne une valeur de pente de 5.8. Ainsi, on peut en conclure que l'algorithme de descente de gradient proximale converge vers la solution du problème avec une vitesse en $O(\frac{1}{k^{5,8}})$. La qualité de l'estimation est confirmée par le coefficient de détermination de la la régression $r = 0.97$, proche de 1.

Temps de calcul et conclusion :

Les mesures suivantes ont été effectuées pour un problème de dimension $n = 200$.

Algorithme	$\gamma = 1$			$\gamma = 10$		
	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-5}$	Convergence	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-5}$	Convergence
Gradient proximal	1,7 s 3042 itér.	12 s 29380 itér.	$O(\frac{1}{k^{1,3}})$	2,1 s 5099 itér.	29 s 67027 itér.	$O(\frac{1}{k^{1,8}})$
Gradient proximal accéléré	0.3 s 1682 itér.	0,7 s 643 itér.	$O(\frac{1}{k^{4,3}})$	0.5 s 1028 itér.	1.8 s 5508 itér.	$O(\frac{1}{k^{2,8}})$
ADMM	1.1 s 1842 itér.	3,2 s 4565 itér.	$O(\frac{1}{k^{5.8}})$	0.1 s 142 itér.	0.1 s 135 itér.	$O(\frac{1}{k^{3.9}})$

À partir de ce tableau comparatif des performances des algorithmes présentés, il apparaît clairement que l'ADMM converge plus rapidement et en moins d'itérations vers la solution que les deux autres algorithmes. Cette rapidité de convergence est due à un exposant n caractéristique de la convergence en $O(\frac{1}{k^n})$ plus élevé que pour les autres algorithmes. L'ADMM est donc à privilégier par rapport aux algorithmes du gradient proximal systématiquement.

3 Applications pour le débruitage d'images

Cette section présente une application de l'ADMM pour débruiter une image à laquelle nous avons ajouté un bruit blanc gaussien de variance σ^2. Elle reprend la section 6 du [3].

3.1 Un problème discret

Dans ce problème, on se place dans un espace $\Omega \subset \Re^{n\times m}$ dans lequel on va discrétiser une image en une matrice $X \in \Omega$ où $X_{i,j}$ correspond à la valeur de l'image au point d'indice et d'ordonnée (ih, jh) de l'image de départ $(1 \leq i \leq n, 1 \leq j \leq m)$. h représente l'espacement entre deux pixels. Dans la suite du problème on prendra $h = 1$. Cette matrice X correspond à une version discrétisée de l'image.

On définit le gradient ∇X de l'image discrétisée par la transformation suivante de $\Omega \longrightarrow \Omega^2$, la méthode des variations totales :

$$(\nabla X)_{i,j} = \begin{pmatrix} (\nabla X)^1_{i,j} \\ (\nabla X)^2_{i,j} \end{pmatrix}$$

Où

$$(\nabla u)^1_{i,j} = \begin{cases} \frac{X_{i+1,j} - X_{i,j}}{h} \text{ si i} < \text{n} \\ 0 \text{ si i} = \text{n} \end{cases}$$

et

$$(\nabla u)^2_{i,j} = \begin{cases} \frac{X_{i,j+1} - X_{i,j}}{h} \text{ si j} < \text{m} \\ 0 \text{ si j} = \text{m} \end{cases}$$

3.2 Le modèle ROF

Le débruitage de l'image se fera avec le modèle proposé par Rudin, Osher et Fatemi appelé modèle ROF. Il se définit par le problème variationnel suivant :

$$\min_X \int_\Omega |DX| + \frac{\gamma}{2}\|X - Y\|_2^2$$

Où $\Omega \subset \Re^{n\times m}$ est le domaine de l'image à d dimensions, $X \in L^1(\Omega)$ est la solution que l'on cherche à approximer et $Y \in L^1(\Omega)$ est l'observation c'est-à-dire l'image bruitée. Le paramètre γ est introduit pour limiter l'overfitting. Le terme $\int_\Omega |DX|$ se simplifie en $\int_\Omega |\nabla X| dx$.

Nous choisissons le modèle ROF pour sa capacité à préserver les discontinuités présentes sans les lisser outre mesure, caractéristique importante dans notre problème de débruitage d'image.

Dans notre cas discret, la fonction de perte que l'on cherche à minimiser se résume à :

$$\min_{X\in\Omega} \|\nabla X\|_1 + \frac{\gamma}{2}\|X - Y\|_2^2$$

avec $\|\nabla u\|_1 = \sum_{i,j} |(\nabla u)_{i,j}|$ et $|(\nabla u)_{i,j}| = \|(\nabla u)^1_{i,j}\|_1 + \|(\nabla u)^1_{i,j}\|_1 = \| \begin{pmatrix} D_v \\ D_h \end{pmatrix} X\|_1$

Où D_v est la matrice des variations verticales et D_h la matrice des variations horizontales. Finalement, nous résolvons le problème suivant :

$$\min_{X\in\Omega} \frac{\gamma}{2}\|X-Y\|_2^2 + \| \begin{pmatrix} D_v \\ D_h \end{pmatrix} X\|_1$$

Ce problème faisant appel à la norme 1 qui est non dérivable, il est nécessaire de le résoudre soit avec un algorithme proximal soit avec l'ADMM. Pour utiliser un algorithme proximal, il est nécessaire d'évaluer l'opérateur proximal de la fonction $X \longmapsto \| \begin{pmatrix} D_v \\ D_h \end{pmatrix} X\|_1$. Cependant, l'opérateur proximal n'admet pas de solution analytique dans notre cas. On peut alors réécrire le problème sous une autre forme qui permettra d'appliquer l'ADMM :

$$\begin{cases} \min\limits_{X\in\Omega} \frac{\gamma}{2}\|X-Y\|_2^2 + \|Z\|_1 \\ \text{s.c } Z - \begin{pmatrix} D_v \\ D_h \end{pmatrix} X = 0 \end{cases}$$

Les itération de l'aglorithme de l'ADMM sont alors les suivantes :

$$X^+ = (\gamma I + \rho \begin{pmatrix} D_v \\ D_h \end{pmatrix}^T \begin{pmatrix} D_v \\ D_h \end{pmatrix})^{-1}(\gamma X - (\begin{pmatrix} D_v \\ D_h \end{pmatrix})^T(U-\rho Z)$$

$$Z^+ = (\begin{pmatrix} D_v \\ D_h \end{pmatrix} X^+ + \frac{(U-\mathbf{1})}{\rho})\mathbb{1}_{(Z>0)} + (\begin{pmatrix} D_v \\ D_h \end{pmatrix} X^+ + \frac{(U+\mathbf{1})}{\rho})\mathbb{1}_{(Z<0)}$$

$$U^+ = U + \rho(\begin{pmatrix} D_v \\ D_h \end{pmatrix} X^+ - Z^+)$$

En appliquant cet algorithme, on obtient les résultats suivants :

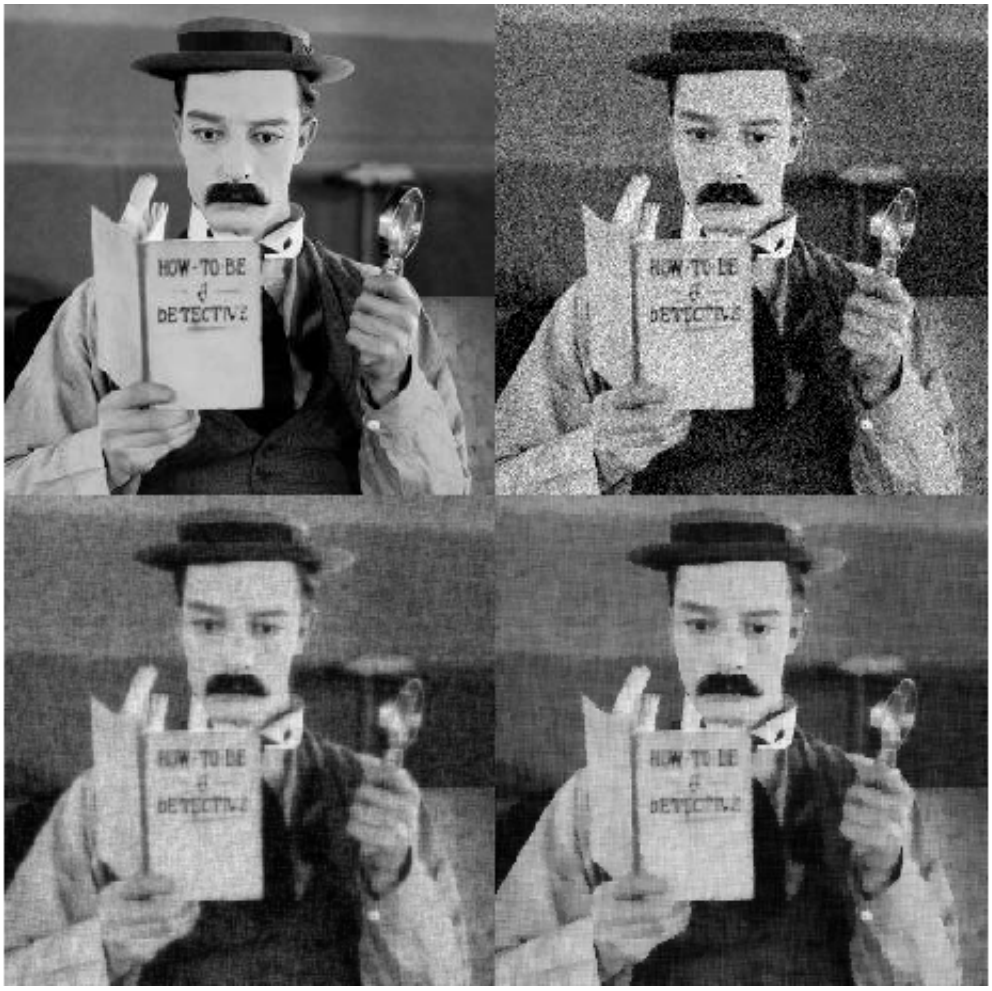

FIGURE 7 – De gauche à droite et de haut en bas : l'image originale, l'image bruitée, l'image débruitée en résolvant le problème : $\gamma\|X - Y\|_2^2 + \| \begin{pmatrix} D_v \\ D_h \end{pmatrix} X\|_2^2$, l'image débruitée en résolvant le problème des variations totales : $\frac{\gamma}{2}\|X - Y\|_2^2 + \| \begin{pmatrix} D_v \\ D_h \end{pmatrix} X\|_1$ avec l'ADMM.

En affichant l'évolution de la valeur de la fonction à minimiser en fonction du nombre d'itération, on constate que l'ADMM permet effectivement de minimiser cette fonction. De plus, on constate qu'au fur et à mesure des itérations, nous nous rapprochons de l'image originale.

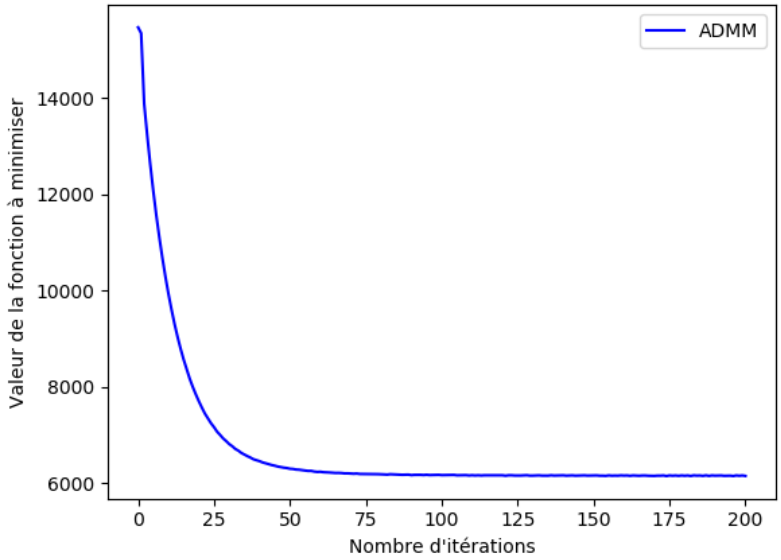

FIGURE 8 – Valeur du modèle ROF en fonction du nombre d'itérations

On peut comparer avec le problème classique de minimisation des variations totales qui est assez proche du modèle ROF puisqu'une norme 2 remplace la norme 1. On obtient alors le problème de minimisation suivant :

$$\min_{X \in \Omega} \gamma \|X - Y\|_2^2 + \| \begin{pmatrix} D_v \\ D_h \end{pmatrix} X \|_2^2$$

Dans le problème de minimisation des variations totales, la fonction à minimiser est pénalisée par le carré des variations totales. Par conséquent, ce modèle a plus tendance à lisser l'image et flouter les contours que le modèle ROF. On en déduit alors que le résultat qu'il donnera sera alors plus éloigné de l'image de départ que le résultat obtenu suite à la minimisation du modèle de ROF. C'est effectivement ce qu'on observe, ainsi en environ 25 itérations, la minimisation du modèle ROF renvoie une image plus proche de l'image originale que le problème de variations totales. On peut donc conclure que le modèle de ROF est plus pertinent pour le débruitage d'images.

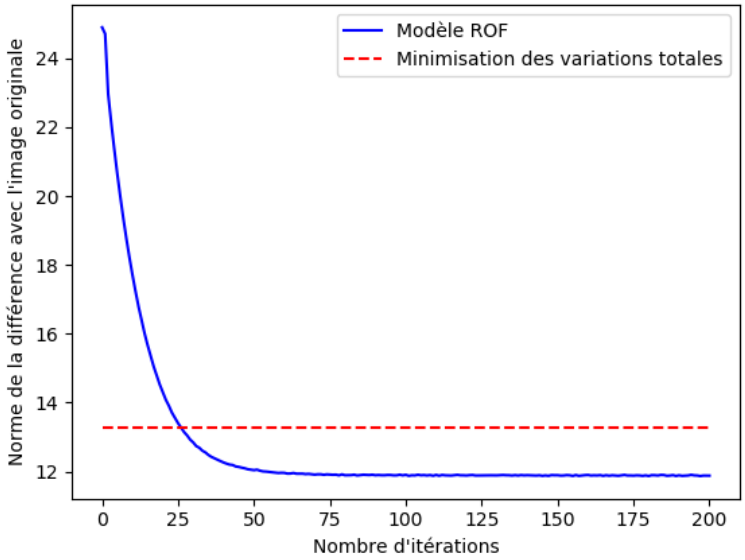

FIGURE 9 – Différence avec l'image originale

On retrouve les caractéristiques de l'ADMM déjà observées à la section précédente : $\log\left(\frac{\|x_0-x_*\|_2^2}{(f(x_k)-f^*)}\right) \geq \log(k)$. De plus, après un régime transitoire de quelques centaines d'itérations, cette suite tend vers une suite affine dont la pente nous donne accès à la valeur de la vitesse de convergence de l'algorithme vers la solution du modèle ROF.

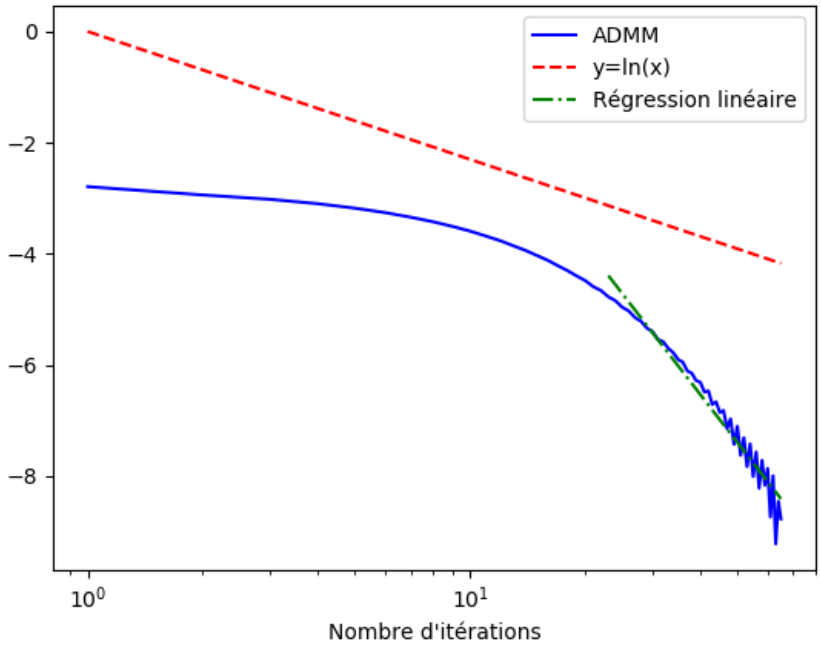

FIGURE 10 – Comparaison entre $\log\left(\frac{\|x_0-x_*\|_2^2}{(f(x_k)-f^*)}\right)$ en vert clair et $\log(k)$ en vert foncé en fonction des itérations

Dans le tableau qui suit, MVT renvoie au problème de Minimisation des Variations Totales.

Variance	$\sigma^2 = 0.01$	$\sigma^2 = 0.05$
Temps pour dépasser la MVT	0.68 s	0.86 s
Nombre d'itération avant de dépasser la MVT	27	32
Écart en norme avec la MVT normalisé par le nombre de pixels de l'image originale	0.052	0.091
Vitesse de convergence	$O(1/k^{3.6})$	$O(1/k^4)$

En conclusion de cette section, l'étude expérimentale menée ici a permis de mettre en évidence l'intérêt d'utiliser le modèle ROF pour débruiter une image plutôt que la minimisation des variations totales. Il s'agit alors d'une utilisation concrète de l'ADMM puisque la fonction à minimiser n'est pas dérivable et que les algorithmes du gradient proximal ne peuvent pas s'appliquer. Cependant, on note tout de même qu'avec l'augmentation de la variance du bruit, la pertinence du recours au modèle de ROF diminue étant donné que l'écart avec la solution au problème de minimisation des variations totales se réduit.

4 Tableau récapitulatif

Type d'analyse	Méthode du gradient	Hypothèses	Vitesse de convergence de $f(x^k) - p^*$ vers 0
Classique	Pas fixe	f convexe, M-lipschitzienne le pas vaut $t \leq \frac{1}{M}$	$O(\frac{1}{n})$
		f fortement convexe et deux fois continûment dérivable le pas $t \in]0, \frac{2}{M}]$.	$O(c^n)$ où $0 < c = 1 - 2t\frac{mM}{m+M} < 1$ avec $0 < t < \frac{2}{m+M}$
	Backtracking	f fortement convexe et deux fois continûment dérivable	$O(c^n)$ où $0 < c = 1 - 2m\alpha \min(1, \frac{\beta}{M}) < 1$ avec $0 < \alpha \leq 0,5$ et $0 < \beta < 1$
	Optimal	f fortement convexe et deux fois continûment dérivable	$O(c^n)$ où $0 < c = 1 - 2\frac{m}{m+M} < 1$
Théorie des opérateurs monotones	Proximal	f et g sont CCP f est M-gradient lipschitz le pas $t \leq \frac{2}{M}$.	$O(\frac{1}{n})$
	Proximal accéléré	f et g sont CCP f est M-gradient lipschitz le pas $t \leq \frac{1}{M}$.	$O(\frac{1}{n^2})$
	ADMM	f et g sont CCP f est M-gradient lipschitz le pas $t \leq \frac{1}{M}$.	$O(\frac{1}{n})$

5 Références

[1] S. Bubeck et al. Convex optimization : Algorithms and complexity. Foundations and Trends® in Machine Learning, 8(3-4) :231–357, 2015.

[2] Randal Douc Sylvain Le Corff, MAT 4506 Machine Learning, Télécom SudParis, 2019.

[3] Antonin Chambolle, Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. 2010.

[4] Ernest K. Ryu, Stephen Boyd, A primer on monotone operator methods, 2016.

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Distributed Optimization and Statistics via Alternating Direction Method of Multipliers, 2010.

[6] Neal Parikh and Stephen Boyd, Proximal Algorithms, Foundations and Trends® in Optimization, Vol. 1, No. 3 :123–231, 2013.

[7] Gabriel Stoltz, Méthodes numériques pour l'optimisation, (CERMICS, Ecole des Ponts Equipe-projet MATHERIALS, INRIA Rocquencourt), 2015.

[8] Stephen Boyd Lieven Vandenberghe, Convex Optimization, Cambridge University Press® :464-613, 2009

[9] Laurent El Ghaoui - Algorithmsforlarge-scaleconvexoptimization—DTU2010, 3.Proximal gradient method