Minimisation de fonctions convexes à gradient lipschitzien

Mathematic Park

3 février 2018

Ce document est constitué des notes que j'ai prises en vue d'un exposé au séminaire de vulgarisation Mathematic Park. Les deux sources principales que j'ai utilisées pour le rédiger sont

— « Introductory lectures on convex optimization : a basic course », de Yurii Nesterov, Springer, 2003 ;

— les notes de cours de Sébastien Bubeck, disponibles sur son blog : https://blogs.princeton.edu/imabandit/orf523-the-complexities-of-optimization/.

1 Introduction

Soit $d \in \mathbb{N}^*$. Soit $f : \mathbb{R}^d \to \mathbb{R}$.

$$\text{Problème 1 :} \quad \min_{x \in \mathbb{R}^d} f(x) = ?$$
$$\text{Problème 2 :} \quad \text{trouver } x_* \in \mathbb{R}^d \text{ tel que } f(x_*) = \min_{x \in \mathbb{R}^d} f(x) \, ?$$

(Dans tout l'exposé, on suppose qu'on ne considère que des fonctions qui ont des minimums, de sorte que ces deux problèmes ont une solution.)
On va s'intéresser au problème 2.

Exemples :

1. (exemple donné par Edoardo Provenzi lors de son exposé à la conférence MIA 2018) couleur à choisir pour remplacer une partie manquante d'une œuvre d'art en minimisant le contraste avec le reste de l'œuvre ($d = 3$: une couleur se représente par un triplet de réels) ;
2. reconstruction de l'image d'un objet à partir d'observations indirectes : ici, x représente une image et $f(x)$ est l'écart entre les observations réalisées sur la vraie image inconnue et les observations qui auraient été réalisées si l'image avait été x ($d \sim 10^4 - 10^6$: le nombre de pixels dans une image) ;

3. choix des paramètres d'un algorithme dont on veut qu'il réalise une certaine tâche : x représente les paramètres et $f(x)$ quantifie l'erreur réalisée par l'algorithme sur la tâche demandée lorsqu'on utilise les paramètres x ($d \sim 1 - 10^7$).

On veut un algorithme qui calcule une approximation du minimiseur.

→ En entrée : f et un réel $\epsilon > 0$.

→ En sortie : x_{approx} tel que

$$f(x_{approx}) \leq f(x_*) + \epsilon.$$

Un algorithme naïf consisterait à évaluer f en un très grand nombre de points et à renvoyer le point où on a obtenu la plus petite valeur. Si f vérifie certaines propriétés simples, cet algorithme fonctionne. Néanmoins, il n'est pas très intéressant en pratique, puisque le nombre de points où il faut d'évaluer f risque d'être très grand.
Ici, nous allons décrire des algorithmes qui permettent de calculer des minimiseurs approchés avec un nombre d'opérations beaucoup plus faible et nous interroger sur le nombre d'opérations minimimal possible.

2 Hypothèses

2.1 Convexité

La première est la plus importante.
Hypothèse 1
La fonction f est convexe : pour tous $x, y \in \mathbb{R}^d$, pour tout $t \in [0; 1]$,

$$f((1-t)x + ty) \leq (1-t)f(x) + tf(y).$$

(Conséquence : si x est un minimum local de f, c'est un minimum global de f.)

2.2 Dérivabilité

Hypothèse 2
La fonction f est dérivable sur $\mathbb{R}^d$: pour tout x, il existe $a_1(x), \ldots, a_n(x) \in \mathbb{R}$ tels que

$$f(x + (h_1, \ldots, h_n)) \approx f(x) + a_1(x)h_1 + \cdots + a_n(x)h_n$$

pour tous $h_1, \ldots, h_n \ll$ petits $\gg$.
Plus formellement :

$$\frac{f(x + (h_1, \ldots, h_n)) - (f(x) + a_1(x)h_1 + \cdots + a_n(x)h_n)}{||(h_1, \ldots, h_n)||} \xrightarrow{h_1, \ldots, h_n \to 0} 0,$$

ce qu'on note $f(x + (h_1, \ldots, h_n)) = f(x) + a_1(x)h_1 + \cdots + a_n(x)h_n + o(||(h_1, \ldots, h_n)||)$.

(La norme est la norme usuelle.)

Définition 2.1. *Pour tout $x \in \mathbb{R}$, on note $\nabla f(x) = (a_1(x), \ldots, a_n(x))$, avec $a_1(x), \ldots, a_n(x)$ les réels définis précédemment. C'est le* gradient *de f au point x.*

Avec cette notation, on a, pour tout $x \in \mathbb{R}^d$,

$$f(x+h) = f(x) + \langle \nabla f(x), h \rangle + o(||h||) \qquad \text{si } h = (h_1, \ldots, h_n).$$

(Le produit scalaire est le produit scalaire usuel.)

2.3 Gradient lipschitzien

Hypothèse 3
Le gradient de f est lipschitzien : il existe $L > 0$ tel que, pour tous $x, y \in \mathbb{R}^d$,

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||.$$

À partir de maintenant, on suppose fixé un L vérifiant cette propriété.

2.4 Conséquence

Lemme 2.2. *Pour tous $x, y \in \mathbb{R}^d$,*

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

Démonstration. Soit $y \in \mathbb{R}^d$ fixé.
Première inégalité
Pour tout $t \in [0; 1]$, d'après notre première hypothèse,

$$(1-t)f(x) + tf(y) \quad \geq \quad f((1-t)x + ty).$$

Le membre de gauche est égal à $f(x) + t(f(y) - f(x))$ et le membre de droite à

$$\begin{aligned} f(x + t(y-x)) &= f(x) + \langle \nabla f(x), t(y-x) \rangle + o(t||y-x||) \\ &= f(x) + t \langle \nabla f(x), y - x \rangle + o(t). \end{aligned}$$

Donc $f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle$.
Deuxième inégalité
Notons $g : t \in [0; 1] \to f((1-t)x + ty)$. C'est une fonction dérivable et

$$g'(t) = \langle \nabla f((1-t)x + ty), y - x \rangle .$$

Donc

$$f(y) = g(1) = g(0)(= f(x)) + \int_0^1 \langle \nabla f((1-t)x + ty), y - x \rangle \, dt.$$

Pour tout $t \in [0; 1]$, par l'hypothèse 3,

$$||\nabla f((1-t)x + ty) - \nabla f(x)|| \leq Lt||x - y||,$$

d'où

$$\langle \nabla f((1-t)x + ty), y - x \rangle \leq \langle \nabla f(x), y - x \rangle + Lt||x - y||^2.$$

Ainsi,

$$\begin{aligned} f(y) &\leq f(x) + \int_0^1 \big(\langle \nabla f(x), y - x \rangle + Lt||x - y||^2 \big) dt \\ &= f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}||x - y||^2. \end{aligned}$$

□

3 Descente de gradient

Notre algorithme va construire une suite $(x_n)_{n\in\mathbb{N}}$ telle que $f(x_n) \overset{n\to+\infty}{\to} \min f$. À chaque étape, x_{n+1} sera défini en fonction de x_n de manière judicieuse, de façon à ce que $f(x_{n+1})$ soit le plus petit possible.
Pour contrôler la vitesse de convergence de $(f(x_n) - \min f)_{n\in\mathbb{N}}$, à chaque étape :

1. en utilisant (1), on majore $f(x_{n+1})$ en fonction de $f(x_n)$ et $\nabla f(x_n)$;
2. en utilisant (1), on minore $\min f$ en fonction de $f(x_n)$ et $\nabla f(x_n)$.

On en déduit une relation de récurrence entre $f(x_n) - f(x_*)$ et $f(x_{n+1}) - f(x_*)$.

3.1 Définition de x_{n+1} et majoration de $f(x_{n+1})$

Soit n fixé.
Pour tout y, par (1),

$$\begin{aligned} f(y) &\leq f(x_n) + \langle \nabla f(x_n), y - x_n \rangle + \frac{L}{2}||x_n - y||^2 \\ &= f(x_n) - \frac{1}{2L}||\nabla f(x_n)||^2 + \frac{L}{2}\left|\left|x_n - \frac{1}{L}\nabla f(x_n) - y\right|\right|^2. \end{aligned}$$

Il est raisonnable de minimiser le membre de droite :

$$x_{n+1} \overset{déf}{=} x_n - \frac{1}{L}\nabla f(x_n).$$

On a alors

$$f\left(x_{n+1}\right) \leq f(x_n) - \frac{1}{2L}||\nabla f(x_n)||^2. \tag{2}$$

3.2 Minoration de $\min f$

On applique (1) à $y = x_*$ (où x_* est un minimiseur de f) :

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

donc

$$\begin{aligned} f(x_n) - \min f \leq \langle \nabla f(x_n), x_n - x_* \rangle &\leq ||\nabla f(x_n)||\,||x_n - x_*||, \\ \Rightarrow \quad \frac{f(x_n) - \min f}{||x_n - x_*||} &\leq ||\nabla f(x_n)||. \end{aligned}$$

En combinant avec (2) :

$$f\left(x_{n+1}\right) - \min f \leq f(x_n) - \min f - \frac{1}{2L}\frac{(f(x_n) - \min f)^2}{||x_n - x_*||^2}.$$

3.3 Vitesse de convergence de la descente de gradient

Théorème 3.1. *Pour tout n,*

$$f(x_n) - \min f \leq \frac{2L||x_0 - x_*||^2}{n}.$$

Démonstration. La suite $(||x_n - x_*||)_{n\in\mathbb{N}}$ est décroissante. En effet, pour tout n,

$$\begin{aligned} ||x_{n+1} - x_*||^2 &= ||x_n - x_*||^2 - \frac{2}{L}\langle \nabla f(x_n), x_n - x_* \rangle + \frac{1}{L^2}||\nabla f(x_n)||^2 \\ &\leq ||x_n - x_*||^2 - \frac{2}{L}(f(x_n) - \min f) + \frac{2}{L}(f(x_n) - f(x_{n+1})) \\ &\leq ||x_n - x_*||^2. \end{aligned}$$

La première inégalité provient du membre gauche de (1) et de l'inégalité (2).

De cette décroissance on déduit, pour tout n,

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

donc

$$\begin{aligned}\frac{1}{f(x_{n+1}) - \min f} &\geq \frac{1}{f(x_n) - \min f} \times \frac{1}{1 - ((f(x_n) - \min f)/(2L||x_0 - x_*||^2))} \\ &\geq \frac{1}{f(x_n) - \min f}\left(1 + \frac{f(x_n) - \min f}{2L||x_0 - x_*||^2}\right) \\ &= \frac{1}{f(x_n) - \min f} + \frac{1}{2L||x_0 - x_*||^2}.\end{aligned}$$

D'où, pour tout $n \in \mathbb{N}^*$,

$$\begin{aligned}\frac{1}{f(x_n) - \min f} &\geq \frac{n}{2L||x_0 - x_*||^2} \\ \Rightarrow \quad f(x_n) - \min f &\leq \frac{2L||x_0 - x_*||^2}{n}.\end{aligned}$$

□

4 Bornes inférieures pour les méthodes de premier ordre

La descente de gradient est un algorithme de premier ordre : pour tout $n \in \mathbb{N}$, x_{n+1} est défini par

$$x_{n+1} = \mathcal{A}_n(x_0, \ldots, x_n, f(x_0), \ldots, f(x_n), \nabla f(x_0), \ldots, \nabla f(x_n)), \tag{3}$$

pour un certain choix de fonctions déterministes $\mathcal{A}_0, \ldots, \mathcal{A}_n, \ldots$.

Théorème 4.1. *Soit $(\mathcal{A}_n)_{n\in\mathbb{N}}$ un ensemble de fonctions définissant un algorithme de premier ordre. Soit $N \leq \frac{d-1}{2}$ fixé. Alors il existe une fonction $f : \mathbb{R}^d \to \mathbb{R}$ convexe, dérivable, de gradient L-lipschitzien telle que, si on définit $(x_n)_{n\in\mathbb{N}}$ par la relation (3) avec $x_0 = 0$,*

$$f(x_N) - \min f \geq \frac{3L||x_0 - x_*||^2}{32(N+1)^2}.$$

Démonstration. On va faire deux hypothèses simplificatrices. La première, qui peut sembler forte, est que, pour tout n,

$$x_{n+1} \in \mathrm{Vect}\{\nabla f(x_0), \ldots, \nabla f(x_n)\}. \tag{4}$$

Il se trouve que cette hypothèse simplifie significativement les définitions mais ne change pas le mécanisme de la preuve.
(Pour voir comment se débarrasser de cette hypothèse, on peut se référer aux notes d'Anatoli Juditsky pour son cours de M2 « Convex optimization, theory and algorithms », disponibles sur Internet.)
La deuxième hypothèse est qu'on suppose

$$N = \frac{d-1}{2}.$$

Si cette hypothèse n'est pas vérifiée, c'est-à-dire si $N < \frac{d-1}{2}$, il suffit de définir $d' = 2N + 1 < d$, de construire une fonction f_0 sur $\mathbb{R}^{d'}$ de la manière qu'on va décrire dans un instant et de la prolonger à $\mathbb{R}^d$ de manière « naïve » en une fonction f. Cela complique légèrement les définitions mais ne change rien au principe de la démonstration.
On définit, pour tout $n \leq d$,

$$E_n = \{(x_1, \dots, x_n, 0, \dots, 0), x_1, \dots, x_n \in \mathbb{R}\} \subset \mathbb{R}^d.$$

On va définir une fonction f vérifiant la propriété suivante :

$$\forall n \in \mathbb{N}, \forall x \in E_n, \qquad \nabla f(x) \in E_{n+1}. \tag{5}$$

Cette propriété implique, en conjonction avec l'hypothèse (4), que, si $x_0 = 0$, alors $x_n \in E_n$ pour tout n. En particulier,

$$f(x_N) \geq \min_{x \in E_N} f(x)$$

et donc

$$f(x_N) - \min f \geq \min_{x \in E_N} f(x) - \min f.$$

Notre but est maintenant de choisir pour f une fonction convexe, dérivable et à gradient lipschitzien, vérifiant la propriété (5), qui soit suffisamment simple pour que $\min_{x \in E_N} f(x) - \min f$ puisse se calculer explicitement.
On va choisir f comme une fonction polynomiale de degré 2 :

$$f(x_1, \dots, x_d) = \text{cte} + \sum_{i=1}^{d} \alpha_i x_i + \sum_{i,j=1}^{d} \beta_{i,j} x_i x_j.$$

On a alors

$$\nabla f(x_1, \dots, x_d) = \begin{pmatrix} \alpha_1 + \sum_j (\beta_{1,j} + \beta_{j,1}) x_j \\ \vdots \\ \alpha_d + \sum_j (\beta_{d,j} + \beta_{j,d}) x_j. \end{pmatrix}$$

Pour que la propriété (5) soit vérifiée, on doit avoir

$$\alpha_2 = \cdots = \alpha_d = 0;$$
$$\forall i, j, |i-j| \geq 2, \beta_{i,j} = 0.$$

Ainsi, (la constante étant indifférente), on peut chercher f sous la forme

$$f(x_1, \ldots, x_d) = \alpha_1 x_1 + \beta_{1,1} x_1^2 + \beta_{1,2} x_1 x_2 + \beta_{2,2} x_2^2 + \beta_{2,3} x_2 x_3 + \ldots.$$

Quitte à faire un changement de variable, on peut (plus ou moins) supposer que tous les $\beta_{i,i}$ sont égaux à une certaine constante β positive :

$$f(x_1, \ldots, x_d) = \alpha_1 x_1 + \beta x_1^2 + \beta_{1,2} x_1 x_2 + \beta x_2^2 + \beta_{2,3} x_2 x_3 + \ldots.$$

On prend $\alpha_1 = -\beta$ et $\beta_{i,i+1} = -\beta$ pour tout i :

$$f(x_1, \ldots, x_d) = \beta \left(-x_1 + x_1^2 - x_1 x_2 + x_2^2 - x_2 x_3 + \ldots \right).$$

Cette fonction est convexe. Pour tout n, on peut de plus calculer son minimum sur E_n ; on obtient en particulier :

$$\min f = -\frac{\beta}{2} \left(1 - \frac{1}{d+1} \right);$$
$$\min_{x \in E_N} f(x) = -\frac{\beta}{2} \left(1 - \frac{1}{N+1} \right).$$

En outre,

$$||x_0 - x_*||^2 \leq \frac{d+1}{3} = \frac{2}{3}(N+1)$$

et on peut vérifier que la fonction f est de gradient 4β-lipschitzien. On prend donc $\beta = \frac{L}{4}$ et on a

$$f(x_N) - \min f \geq \frac{L}{8} \left(\frac{1}{N+1} - \frac{1}{d+1} \right) = \frac{L}{16(N+1)} \geq \frac{3L}{32} \frac{||x_0 - x_*||^2}{(N+1)^2}.$$

□

Remarque 4.2. *Le théorème qui précède montre que la convergence ne peut pas être plus rapide que* $O\left(\frac{1}{(N+1)^2}\right)$ *si on autorise la dimension* d *à être arbitrairement grande.*
Si, au contraire, on suppose que d *est fixée, le théorème ne dit rien sur la précision minimale possible au-delà des* $\frac{d-1}{2}$ *premières itérations. Et de fait, en dimension fixée, il existe des algorithmes de premier ordre à convergence plus rapide que* $O\left(\frac{1}{(N+1)^2}\right)$ *(voir la méthode des « centers of gravity »).*

5 Gradient accéléré

La vitesse de convergence de la descente de gradient est en $O\left(\frac{1}{N}\right)$ alors que notre borne inférieure est en $O\left(\frac{1}{N^2}\right)$. Cela suggère que la borne inférieure n'est pas optimale ou qu'il existe des méthodes de premier ordre plus rapides que la descente de gradient. C'est la deuxième hypothèse qui est la bonne. Dans le reste de l'exposé, nous allons décrire une méthode de premier ordre de précision optimale (à la constante multiplicative près), due à Yurii Nesterov.
Rappel sur la descente de gradient : on définit $(x_n)_{n\in\mathbb{N}}$ par

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

et, à chaque étape n,

1. on majore $f(x_{n+1})$ en fonction de $f(x_n)$ et $\nabla f(x_n)$;
2. on minore $\min f$ en fonction de $f(x_n)$ et $\nabla f(x_n)$.

Deux erreurs de conception :

1. À l'étape n, le point où on calcule le gradient est x_n, notre approximation de la solution ; il y a peut-être un point où le gradient est plus informatif.
   Remède : notre nouvel algorithme va calculer simultanément deux suites, $(x_n)_{n\in\mathbb{N}}$ et $(y_n)_{n\in\mathbb{N}}$. La première sera la suite des approximations et la deuxième la suite des points où on va calculer le gradient.
2. À l'étape n, on majore $f(x_{n+1})$ et (surtout) on minore $\min f$ en fonction de $f(x_n)$ et $\nabla f(x_n)$ seulement. On ignore l'information qui provient de $x_0, \dots, x_{n-1}$.

5.1 Minoration plus sophistiquée de $\min f$

On note x_* un point tel que $f(x_*) = \min f$. On suppose $x_0 = 0$ et donc $||x_0 - x_*|| = ||x_*||$.
Soit n fixé. D'après (1), pour tout $k = 0, \dots, n$,

$$\min f = f(x_*) \geq f(y_k) + \langle \nabla f(y_k), x_* - y_k \rangle .$$

Imaginons que $\alpha_0^{(n)}, \dots, \alpha_n^{(n)}$ sont des réels positifs fixés tels que $\sum_{k=0}^n \alpha_k^{(n)} = 1$. Alors

$$\begin{aligned}
\min f &\geq \sum_{k=0}^{n} \alpha_k^{(n)} \left(f(y_k) + \langle \nabla f(y_k), x_* - y_k \rangle \right) \\
&= \sum_{k=0}^{n} \alpha_k^{(n)} \left(f(y_k) - \langle \nabla f(y_k), y_k \rangle \right) + \left\langle \sum_{k=0}^{n} \alpha_k^{(n)} \nabla f(y_k), x_* \right\rangle .
\end{aligned}$$

Notons $S_n = \sum_{k=0}^{n} \alpha_k^{(n)} \nabla f(y_k)$ et imaginons que ϵ_n est un réel strictement positif fixé. Alors

$$\begin{aligned}
\min f &\geq \sum_{k=0}^{n} \alpha_k^{(n)} \left(f(y_k) - \langle \nabla f(y_k), y_k \rangle \right) + \left\langle \frac{1}{\epsilon_n} S_n, \epsilon_n x_* \right\rangle \\
&\geq \sum_{k=0}^{n} \alpha_k^{(n)} \left(f(y_k) - \langle \nabla f(y_k), y_k \rangle \right) - \left|\left| \frac{1}{\epsilon_n} S_n \right|\right| \, ||\epsilon_n x_*|| \\
&\geq \sum_{k=0}^{n} \alpha_k^{(n)} \left(f(y_k) - \langle \nabla f(y_k), y_k \rangle \right) - \frac{1}{2\epsilon_n^2} ||S_n||^2 - \frac{1}{2} \epsilon_n^2 ||x_*||^2.
\end{aligned}$$

Notons m_n ce dernier minorant.

5.2 Définition de $(x_n)_{n\in\mathbb{N}}$ et $(y_n)_{n\in\mathbb{N}}$

Supposons n fixé. Comment définir x_{n+1} et y_{n+1} à partir de $x_0, \ldots, x_n$ et $y_0, \ldots, y_n$?
Pour x_{n+1}, on va garder la définition de la descente de gradient, en remplaçant x_n par y_{n+1} :

$$x_{n+1} = y_{n+1} - \frac{1}{L} \nabla f(y_{n+1}).$$

Comment définir y_{n+1} ? On veut pouvoir contrôler

$$f(x_{n+1}) - \min(f) \leq f(x_{n+1}) - m_{n+1}.$$

Peut-on choisir y_{n+1} de façon à ce qu'il soit possible d'établir une relation de récurrence entre $f(x_{n+1}) - m_{n+1}$ et $f(x_n) - m_n$? Par exemple, on aimerait choisir y_{n+1} de façon à ce qu'à coup sûr,

$$f(x_{n+1}) - m_{n+1} \leq \lambda_{n+1}(f(x_n) - m_n),$$

pour un certain $\lambda_{n+1} \in [0; 1[$.
Insérons la définition de m_n et m_{n+1} dans cette inégalité. On veut :

$$\begin{aligned}
f(x_{n+1}) - \sum_{k=0}^{n+1} \alpha_k^{(n+1)} \left(f(y_k) - \langle \nabla f(y_k), y_k \rangle \right) + \frac{1}{2\epsilon_{n+1}^2} ||S_{n+1}||^2 + \frac{1}{2} \epsilon_{n+1}^2 ||x_*||^2 \\
\leq \lambda_{n+1} \left(f(x_n) - \sum_{k=0}^{n} \alpha_k^{(n)} \left(f(y_k) - \langle \nabla f(y_k), y_k \rangle \right) + \frac{1}{2\epsilon_n^2} ||S_n||^2 + \frac{1}{2} \epsilon_n^2 ||x_*||^2 \right).
\end{aligned}$$

Imposons

$$\alpha_0^{(n+1)} = \lambda_{n+1} \alpha_0^{(n)}, \ldots, \alpha_n^{(n+1)} = \lambda_{n+1} \alpha_n^{(n)}, \alpha_{n+1}^{(n+1)} = 1 - \lambda_{n+1}$$
$$\epsilon_{n+1} = \sqrt{\lambda_{n+1}} \epsilon_n.$$

On ne veut alors plus que

$$f(x_{n+1}) - (1-\lambda_{n+1})\left(f(y_{n+1}) - \langle \nabla f(y_{n+1}), y_{n+1}\rangle\right) + \frac{1}{2\epsilon_{n+1}^2}\left|\left|S_{n+1}\right|\right|^2 \\ \leq \lambda_{n+1}\left(f(x_n) + \frac{1}{2\epsilon_n^2}\left|\left|S_n\right|\right|^2\right).$$

En utilisant les inégalités

$$f(x_{n+1}) \leq f(y_{n+1}) - \frac{1}{2L}||\nabla f(y_{n+1})||^2$$
$$\text{et} \quad f(y_{n+1}) \leq f(x_n) - \langle \nabla f(y_{n+1}), x_n - y_{n+1}\rangle,$$

on n'a plus qu'à obtenir

$$-\frac{1}{2L}||\nabla f(y_{n+1})||^2 - \lambda_{n+1}\langle \nabla f(y_{n+1}), x_n - y_{n+1}\rangle + (1-\lambda_{n+1})\langle \nabla f(y_{n+1}), y_{n+1}\rangle \\ + \frac{1}{2\epsilon_{n+1}^2}\left|\left|S_{n+1}\right|\right|^2 \leq \frac{\lambda_{n+1}}{2\epsilon_n^2}\left|\left|S_n\right|\right|^2$$
$$\iff \quad -\frac{1}{2L}||\nabla f(y_{n+1})||^2 - \lambda_{n+1}\langle \nabla f(y_{n+1}), x_n\rangle + \langle \nabla f(y_{n+1}), y_{n+1}\rangle \\ + \frac{1}{2\epsilon_{n+1}^2}\left|\left|S_{n+1}\right|\right|^2 \leq \frac{\lambda_{n+1}}{2\epsilon_n^2}\left|\left|S_n\right|\right|^2.$$

Puisque

$$\begin{aligned} S_{n+1} &= \sum_{k=0}^{n+1} \alpha_k^{(n+1)} \nabla f(y_k) \\ &= \lambda_{n+1}\left(\sum_{k=0}^{n} \alpha_k^{(n)} \nabla f(y_k)\right) + (1-\lambda_{n+1})\nabla f(y_{n+1}) \\ &= \lambda_{n+1} S_n + (1-\lambda_{n+1})\nabla f(y_{n+1}), \end{aligned}$$

cette dernière inégalité est équivalente à

$$-\frac{1}{2L}||\nabla f(y_{n+1})||^2 - \lambda_{n+1}\langle \nabla f(y_{n+1}), x_n\rangle + \langle \nabla f(y_{n+1}), y_{n+1}\rangle \\ + \frac{1}{2\epsilon_n^2}\left(2(1-\lambda_{n+1})\langle S_n, \nabla f(y_{n+1})\rangle + \frac{(1-\lambda_{n+1})^2}{\lambda_{n+1}}||\nabla f(y_{n+1})||^2\right) \leq 0,$$

c'est-à-dire

$$\left(\frac{(1-\lambda_{n+1})^2}{2\lambda_{n+1}\epsilon_n^2} - \frac{1}{2L}\right)||\nabla f(y_{n+1})||^2 + \left\langle \frac{1-\lambda_{n+1}}{\epsilon_n^2}S_n - \lambda_{n+1}x_n + y_{n+1}, \nabla f(y_{n+1})\right\rangle \leq 0.$$

Il suffit donc de choisir λ_{n+1} tel que

$$\frac{(1-\lambda_{n+1})^2}{2\lambda_{n+1}\epsilon_n^2} - \frac{1}{2L} = 0 \qquad \overset{(\text{car } \lambda_{n+1}\in[0;1])}{\iff} \qquad \lambda_{n+1} = 1 + \frac{\epsilon_n^2}{2L} - \sqrt{\frac{\epsilon_n^2}{2L} + \frac{\epsilon_n^4}{4L^2}}$$

et

$$y_{n+1} = \lambda_{n+1}x_n - \frac{1-\lambda_{n+1}}{\epsilon_n^2}S_n.$$

Quelques manipulations algébriques (voir la dernière page pour plus de détails) permettent de voir que l'algorithme qu'on vient d'obtenir est le suivant.

Data: La fonction f, un entier N.
On pose $\lambda_0 = 0$ et $y_0 = x_0 = 0$ (et, implicitement, $\epsilon_0 = \sqrt{L}$)
for $n = 1$ *to* N **do**
  $\lambda_n = 1 - \frac{(1-\lambda_{n-1})^2}{2}\left(\sqrt{1 + \frac{4}{(1-\lambda_{n-1})^2}} - 1\right)$;
  $y_n = x_{n-1} + \frac{\lambda_{n-1}(1-\lambda_n)}{1-\lambda_{n-1}}(x_{n-1} - x_{n-2})$;
    (sauf si $n = 1$, auquel cas on pose $y_1 = -\frac{1-\lambda_1}{L}\nabla f(0)$)
  $x_n = y_n - \frac{1}{L}\nabla f(y_n)$;
end

Algorithm 1: Descente de gradient accélérée

Théorème 5.1. *La suite $(x_n)_{n\in\mathbb{N}}$ renvoyée par l'algorithme qu'on vient d'obtenir vérifie, pour tout n,*

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

Démonstration. L'algorithme est construit pour garantir, pour tout $n \in \mathbb{N}$,

$$f(x_{n+1}) - m_{n+1} \leq \lambda_{n+1}(f(x_n) - m_n).$$

On a donc pour tout $n \in \mathbb{N}$

$$\begin{aligned} f(x_n) - \min f &\leq f(x_n) - m_n \\ &\leq \lambda_1 \dots \lambda_n (f(x_0) - m_0) \\ &= \lambda_1 \dots \lambda_n \left(\frac{1}{2L}||\nabla f(0)||^2 + \frac{L}{2}||x_*||^2\right) \\ &\leq \lambda_1 \dots \lambda_n L||x_*||^2. \end{aligned}$$

La dernière inégalité utilise la lipschitziannité du gradient et le fait que $\nabla f(x_*) = 0$.
Si on pose, pour tout n, $\mu_n = \frac{1}{1-\lambda_n}$, la formule qui définit λ_n devient

$$\mu_n = \frac{1 + \sqrt{1 + 4\mu_{n-1}^2}}{2} \geq \mu_{n-1} + \frac{1}{2}.$$

On en déduit par récurrence que, pour tout n, $\mu_n \geq 1 + \frac{n}{2}$. Ainsi, pour tout n,

$$\lambda_n \geq \frac{n}{n+2}.$$

D'où

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

□

Remarque 5.2. *L'algorithme que nous venons de voir est facile à implémenter et, comme le garantit le théorème précédent, sa précision est optimale (à constante multiplicative près) parmi l'ensemble des méthodes de premier ordre. Il présente néanmoins l'inconvénient d'être peu intuitif. En donner une interprétation « éclairante » est une question de recherche qui, à ma connaissance, n'a toujours pas reçu, à ce jour, de réponse totalement satisfaisante.*

A Simplification de la formule de mise à jour de y_n

D'après l'égalité

$$\frac{(1-\lambda_{n+1})^2}{2\lambda_{n+1}\epsilon_n^2} - \frac{1}{2L} = 0,$$

on a

$$y_{n+1} = \lambda_{n+1}x_n - \frac{\lambda_{n+1}}{(1-\lambda_{n+1})L}S_n$$

et, de même,

$$y_n = \lambda_n x_{n-1} - \frac{\lambda_n}{(1-\lambda_n)L}S_{n-1}.$$

En outre,

$$S_n = \lambda_n S_{n-1} + (1-\lambda_n)\nabla f(y_n).$$

Combiné avec l'égalité $\lambda_n S_{n-1} = (1-\lambda_n)L(\lambda_n x_{n-1} - y_n)$ qui vient de la formule d'avant, cela donne

$$S_n = (1-\lambda_n)L(\lambda_n x_{n-1} - y_n) + (1-\lambda_n)\nabla f(y_n).$$

Ainsi,

$$\begin{aligned}
y_{n+1} &= \lambda_{n+1}x_n - \frac{\lambda_{n+1}}{(1-\lambda_{n+1})L}\left((1-\lambda_n)L(\lambda_n x_{n-1} - y_n) + (1-\lambda_n)\nabla f(y_n)\right) \\
&= \lambda_{n+1}x_n - \frac{\lambda_{n+1}\lambda_n(1-\lambda_n)}{1-\lambda_{n+1}}x_{n-1} + \frac{\lambda_{n+1}}{1-\lambda_{n+1}}(1-\lambda_n)\left(y_n - \frac{1}{L}\nabla f(y_n)\right) \\
&= \lambda_{n+1}x_n - \frac{\lambda_{n+1}\lambda_n(1-\lambda_n)}{1-\lambda_{n+1}}x_{n-1} + \frac{\lambda_{n+1}}{1-\lambda_{n+1}}(1-\lambda_n)x_n \\
&= x_n + \left(\lambda_{n+1} - 1 + \frac{1-\lambda_n}{1-\lambda_{n+1}}\lambda_{n+1}\right)x_n - \frac{\lambda_{n+1}\lambda_n(1-\lambda_n)}{1-\lambda_{n+1}}x_{n-1} \\
&= x_n + \frac{\lambda_n(1-\lambda_{n+1})}{1-\lambda_n}(x_n - x_{n-1}).
\end{aligned}$$

On a utilisé le fait que

$$\frac{(1-\lambda_{n+1})^2}{\lambda_{n+1}} = \frac{\epsilon_n^2}{L} = \lambda_n\frac{\epsilon_{n-1}^2}{L} = (1-\lambda_n)^2.$$