Méthodes de descente: l'algorithme du gradient

1. Contenu

Le **plan** de cette session est le suivant:

- Un rappel en image de la convexité, minimun local / global,
- Lien entre minimisation et système linéaire dans le cas quadratique,
- Les algorithmes de descente en général,
- L'algorithme du gradient à pas optimal en particulier.

2. Optimisation: fonctionnelles convexes, minimum local / global

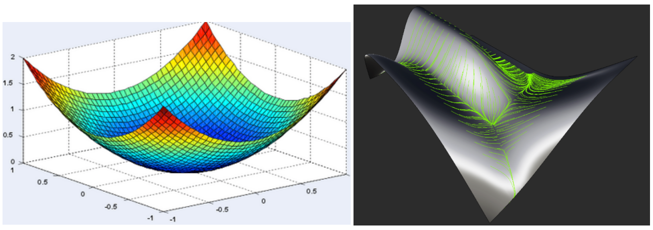
On s'attache ici à vouloir approcher numériquement la solution du problème suivant:

$$\boxed{\left\{ \begin{array}{c} \text{Trouver } x^* \in \mathbb{R}^n \text{ tel que:} \\ j(x^*) = \min_{x \in \mathbb{R}^n} j(x) \end{array} \right.}$$

où j est une fonction définie de $\mathbb{R}^n$ à valeurs dans $\mathbb{R}$ (on dit alors que j est une *fonctionnelle* définie sur $\mathbb{R}^n$). On suppose que j est régulière dans le sens où j est de classe C^1 au moins (et C^2 si besoin).

Les problèmes d'optimisation sont omniprésents en ingénierie et constituent d'ailleurs souvent un des challenges en termes d'innovation ou compétivité. Aussi, bien des phénomènes physiques à l'équilibre (en mécanique des fluides, structures, neutronique, ondes etc etc) peuvent être décrit par une telle formulation d'optimisation / minimisation.

L'existence et l'éventuelle unicité de minima locaux ou minimum global repose sur les propriètés de convexité de la fonctionnelle j, et aussi de l'ensemble dans lequel on recherche la / les solutions. Voici deux exemples qui illustrent dans le cas bidimensionnel ($n = 2$) une fonction quadratique strictement convexe (le cas "idéal"), et une fonctionnelle moins triviale comportant plusieurs minima locaux.



Source image: "madeincalifornia.com"

Rappelons que la condition nécessaire d'extremum (local) s'écrit (hors du bord éventuel de l'espace de recherche):

$$\boxed{\nabla j(x) = 0}$$

Aussi rappelons que la distinction entre minimum et maximum local (voire un point selle) peut se faire sur la base des propriétés de la Hessienne de j en ces points.

Dans le cas j strictement convexe, la condition nécessaire ci-dessus devient une condition suffisante.

Nous vous renvoyons à votre cours favori d'analyse mathématique pour ces questions d'extrema locaux.

3. Lien minimisation - système linéaire

Montrons le lien qui existe entre un système linéaire $Ax = b$ avec A *matrice symétrique définie positive* et la minimisation de la *fonctionnelle quadratique associée.*

Ce lien qui peut généralement s'interpréter d'un point de vue physique (la fonctionnelle j correspond alors à l'*énergie du système* à l'équilibre décrit/modélisé par le système linéaire).

Ce lien ouvre la possibilité de résoudre un même problème par des approches numériques différentes (résolution directe du système linéaire ou minimisation d'une fonctionnelle).

Proposition. *Soit la fonctionelle j définie de $\mathbb{R}^n$ dans $\mathbb{R}$ par: $j(x) = \frac{1}{2}(Ax, x) - (b, x)$ avec A matrice symétrique définie positive. Alors le problème d'optimisation précédent admet une unique solution $x^* \in \mathbb{R}^n$, et cette solution vérifie le système linéaire: $Ax = b$.*

Proof. Développons l'expression de j:

$$j(x) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_ix_j - \sum_{i=1}^{n} b_ix_i$$

Par définition, la fonctionnelle j est quadratique donc strictement convexe, cf figure. Elle admet donc un unique minimum global dans $\mathbb{R}^n$.
Ce minimum est caractérisé par la condition nécessaire (qui est ici également suffisante): $\nabla j(x) = 0$. Or on a: $\forall i \in \{1, .., n\}, \partial_i j(x) = \sum_{j=1}^{n} a_{ij}x_j - b_ix_i = (Ax)_i$, ce qui montre le résultat. $\square$

Notons aussi que la Hessienne de j est constante et vaut: $D^2j(x) = A,\ \forall x \in \mathbb{R}^n$.

Cette proposition est à la base de la méthode de résolution de systèmes linéaires de référence pour A matrice symétrique définie positive, A grande et creuse: *la méthode du Gradient Conjugué.*
Cette méthode du GC est une méthode de descente semblable à la méthode du gradient que nous allons développer.

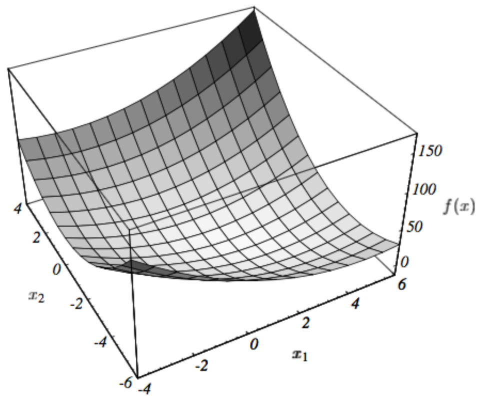


Figure 2: Graph of a quadratic form $f(x)$. The minimum point of this surface is the solution to $Ax = b$.

4. Algorithmes de descente: principe

Le présent problème de minimisation pour une fonctionnelle j régulière peut naturellement se résoudre par une méthode dite de descente.

Formalisme des algorithmes de descente. Il s'agit d'algorithmes itératifs. Partant d'un point initial x_0 (le mieux choisi possible ...), on définit un nouvel itéré $x^{(n+1)}$ qui vérifie simplement:

$$j(x^{(n+1)}) < j(x^{(n)})$$

Pour cela on doit déterminer:

- la *direction de descente* $d^{(n)}$,
- le *pas de descente* $\alpha^{(n)}$ (jusqu'où descend-on ?)

Et on pose:

$$\boxed{x^{(n+1)} = x^{(n)} + \alpha^{(n)}d^{(n)}\ ,\quad n \geq 0}$$

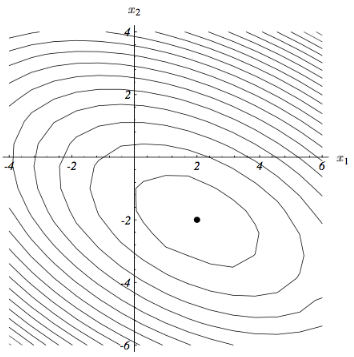


Figure 3: Contours of the quadratic form. Each ellipsoidal curve has constant $f(x)$.

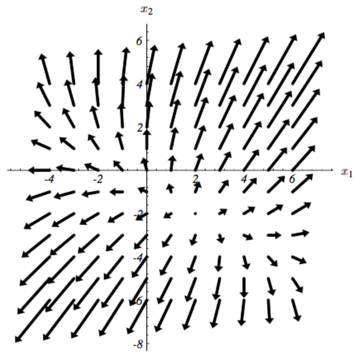


Figure 4: Gradient $f'(x)$ of the quadratic form. For every x, the gradient points in the direction of steepest increase of $f(x)$, and is orthogonal to the contour lines.

5. Algorithmes de descente: principe (2)

La détermination du pas de descente s'appelle la *recherche linéaire*.
La direction de descente d est basée sur l'information du gradient $\nabla j(x)$, cf figure.
A partir d'un point x, une direction de descente d est telle que: $< \nabla j(x), d >< 0$;
tandis que le pas de descente est nécessairement positif.
$< .,. >$ désigne le produit scalaire dans $\mathbb{R}^n$.

En effet, écrivons le développement limité de Taylor au point courant $x^{(n)}$, à l'ordre 2, dans la direction d_n. On a:

$$j(x^{(n+1)}) = j(x^{(n)}) + < \nabla j(x^{(n)}), x^{(n+1)} - x^{(n)} > + < D^2 j(x^{(n)} \cdot (x^{(n+1)} - x^{(n)}), (x^{(n+1)} - x^{(n)}) > + \mathcal{O}(\|x^{(n+1)} - x^{(n)}\|^3)$$

Ce qui donne:

$$\boxed{j(x^{(n+1)}) - j(x^{(n)}) = \alpha^{(n)} < \nabla j(x^{(n)}), d^{(n)} > + \mathcal{O}(\|x^{(n+1)} - x^{(n)}\|^2)}$$

Donc pour d_n direction de descente non nulle i.e. tel que $< \nabla j(x), d >< 0$
et α_n *strictement positif,* $\alpha_n > 0$, suffisamment petit,
on obtient: $j(x^{(n+1)}) < j(x^{(n)})$.

Soit la décroissance de j .

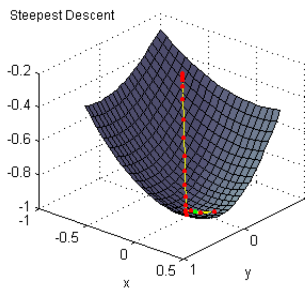
6. Algorithmes de descente: convergence et arrêt

A propos du choix du point initial $x^{(0)}$.

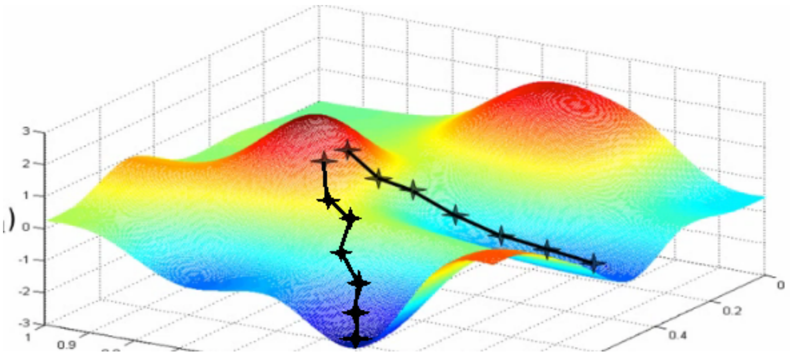
Tout d'abord, comme dans la majorité des algorithmes itératifs, il est judicieux de choisir le point initial le plus proche possible de la solution, à partir d'a-priori que l'on peut avoir, connaissant le problème traité.

Dans le cas d'un méthode de descente, ce choix peut s'avérer être particulièrement crucial.

En effet, si j est strictement convexe (e.g. la fonctionnelle quadratique précédente), pas de problème, tous les chemins de descente mènent à l'unique minimum global !



Par contre pour une fonctionnelle qui admet plusieurs minima locaux, le fait de descendre uniquement, conduit au minimum local du "bassin versant courant"...



Critères d'arrêt. Un algorithme de descente va faire diminuer le critère / la fonctionnelle j, et va converger vers un minimum local x^* (voire un minimum global si j strictement convexe).
Comment savoir si l'algorithme à bien convergé ?

En observant notamment l'évolution des deux critères suivants:

$$\|\nabla j(x^{(n+1)}))\| < \varepsilon_1 \text{ et } \frac{j(x^{(n+1)}) - j(x^{(n)})}{j(x^{(n)})} < \varepsilon_2$$

7. L'algorithme du gradient à pas optimal

L'algorithme du gradient consiste à descendre selon la direction de la plus profonde descente (d'où son autre terminologie: *l'algorithme de la plus profonde descente*).

C'est à dire que l'on pose:

$$\boxed{d^{(n)} = -\nabla j(x^{(n)}) \ , \quad \forall n \geq 0}$$

Un choix classique du pas des descente α consiste à choisir le *pas optimal*,

à savoir le scalaire strictement positif qui minimise le critère j *selon la direction de descente* d, soit:

$$\boxed{\alpha^{(n)} \in \mathbb{R}^+ \text{ tel que: } j(x^{(n)} + \alpha^{(n)} d^{(n)}) = \min_{\alpha \in \mathbb{R}^+} j(x^{(n)} + \alpha d^{(n)})}$$

Dans des cas particuliers, et notamment dans le cas à venir où j est la forme quadratique associée à une matrice symétrique définie positive A, on peut facilement calculer explicitement la valeur du pas optimal.

Les graphiques suivants vous illustrent les différents propos.

Scale document down

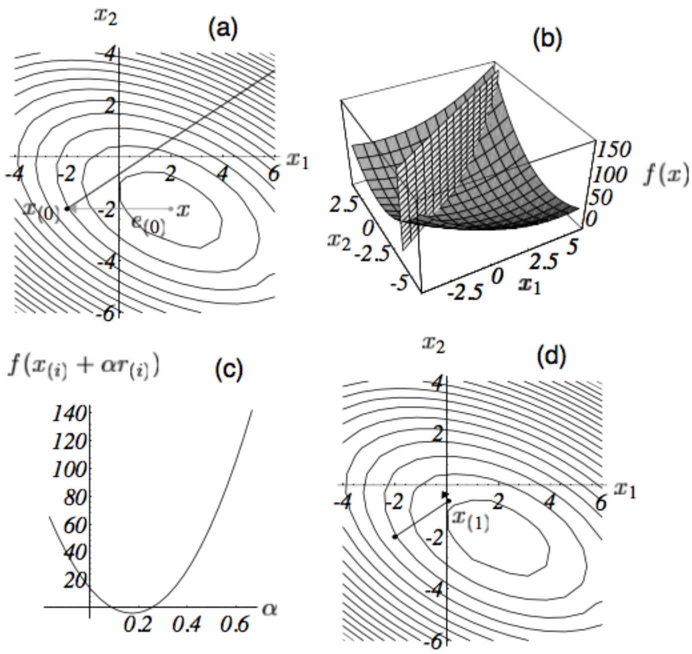


Figure 6: The method of Steepest Descent. (a) Starting at $[-2, -2]^T$, take a step in the direction of steepest descent of f. (b) Find the point on the intersection of these two surfaces that minimizes f. (c) This parabola is the intersection of surfaces. The bottommost point is our target. (d) The gradient at the bottommost point is orthogonal to the gradient of the previous step.

Source image: J.R. Shewchuk, Carnegie Mellon Univ., 1994.

8. L'algorithme du gradient: convergence

On peut montrer le résultat suivant.

Theorem. Soit A une matrice *symétrique définie positive*, et j définie par: $j(x) = \frac{1}{2}(Ax, x) - (b, x)$ (j ressemble à la paraboloïde vue précédemment). Alors la méthode du gradient à pas optimal appliquée à j converge quelque soit le point initial x_0.

De plus , on a l'estimation suivante:

$$\|x^{(n+1)} - x^*\|_A \leq \left(\frac{cond_2(A) - 1}{cond_2(A) + 1} \right)^n \|x^{(0)} - x^*\|_A$$

où $\|.\|_A$ *est la norme associée à* A *et* $cond_2(A)$ *est le conditionnement euclidien de* A.

Rappels.
- La norme associée à A est définie ainsi: $\|x\|_A^2 = (Ax, x)$.
- Le conditionnement euclidien de A, $cond_2(A)$, est défini ainsi: $cond_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$.

Ce théorème montre la convergence inconditionnelle de l'algorithme du gradient pour j fonctionnelle *quadratique.*

Aussi, cette estimation montre que la vitesse de convergence dépend pleinement du *conditionnement* de la matrice A.

Typiquement si $cond_2(A) = 1$ i.e. la matrice A est unitaire, alors la méthode converge en une seule itération (dans le cas $n = 2$, le graphe de j serait à symétrie de révolution).

Inversement si $cond_2(A) >>$ (fréquent en pratique) (dans le cas $n = 2$, la paraboloïde serait très applatie...), alors la méthode va converger extrêmement lentement...

Cette remarque nous montre que les propriétés de conditionnement du système traité sont déterminantes d'où ensuite la construction de méthodes plus sophistiquées dites *préconditionnées* (e.g. la méthode du *gradient conjugué préconditionné).*

9. L'algorithme du gradient: convergence (2)

Enfin, on peut montrer la

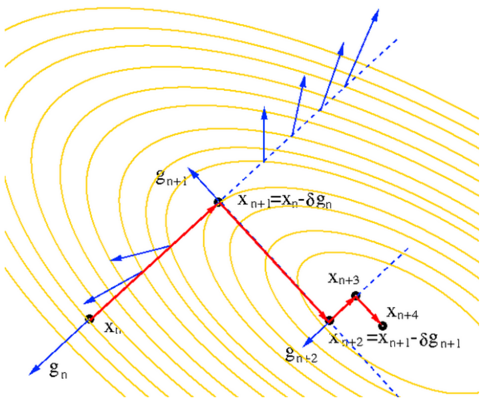
Proposition. Soit A une matrice *symétrique définie positive* et j définie par: $j(x) = \frac{1}{2}(Ax, x) - (b, x)$.

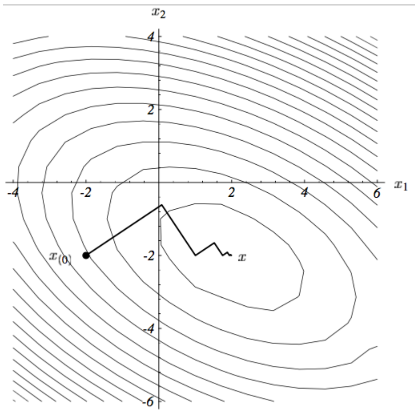
Alors *à chaque itération* de la méthode du gradient à pas optimal, *on a:*

$$d_n \perp d_{n-1}$$

i.e. deux directions de descente successives sont orthogonales.

Autrement dit on descend en *zig-zag* ! Les figures illustrent ces propos.





Cette dernière propriété porte en fait le coup de grâce à cet algorithme du gradient à pas optimal, algorithme qui ne doit *a-priori pas* être utilisé en pratique car bien trop lent à converger !...

Par contre, cet algorithme reste à la base des méthodes de descente de référence (e.g. gradient conjugué, quasi-Newton BFGS).

Typiquement, la *méthode du gradient conjugué* (avec préconditionnement éventuel) est la méthode de référence de résolution du système $Ax = b$ pour A symétrique définie positive, grande ($n >> 10^3$) et creuse (bien moins que 1/100 d'éléments non nuls).