Méthodes numériques pour l'optimisation

Gabriel STOLTZ

stoltz@cermics.enpc.fr

(CERMICS, Ecole des Ponts & Equipe-projet MATHERIALS, INRIA Rocquencourt)

Calcul scientifique, Ecole des Ponts, 12 mars 2015

Méthode de gradient (1)

Problème d'optimisation sans contrainte

$$\inf_{v \in V} J(v), \qquad V \text{ Hilbert}$$

- **Dimension finie :** discrétisation du problème sur une base idoine, fonctionnelle J différentiable

- **Objectif :** construire un point critique de manière itérative

$$v^k \to v \qquad \text{où} \qquad \nabla J(v) = 0$$

- **Principe :** pour $v^k \in V$ donné, direction de descente $d^k = -\nabla J(v^k)$:

$$\text{pour } t \text{ suffisamment petit,} \qquad J(v^k + td^k) \leqslant J(v^k)$$

Méthode de gradient (2)

- **Initialisation**
  - choisir $v^0 \in V$ et poser $k := 0$
  - choisir le pas $\lambda > 0$
  - fixer un seuil de convergence $\varepsilon > 0$

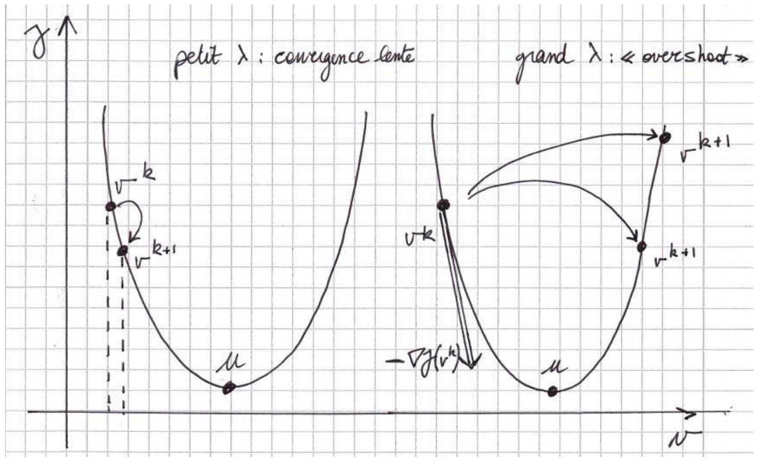
- **Itérations** (boucle sur k)
  - calculer $\nabla J(v^k)$
  - choisir comme direction de descente $d^k = -\nabla J(v^k)$
  - déterminer v^{k+1} selon la formule

$$v^{k+1} = v^k + \lambda d^k$$

  - test de convergence : $\dfrac{\|v^{k+1} - v^k\|_V}{\|v^0\|_V} \leqslant \varepsilon$ ou $\dfrac{|J(v^{k+1}) - J(v^k)|}{J(v^0)} \leqslant \varepsilon$

- Méthode de **gradient à pas fixe** : choix de λ ? convergence ?

Méthode de gradient (3)



- La convergence est très lente si λ est trop petit

- Si λ est trop grand, on peut ne pas converger !

Convergence de la méthode de gradient (1)

- Méthode de gradient à pas fixe = itération de point fixe sur

$$J_\lambda(v) = v - \lambda \nabla J(v)$$

En effet, $v^{k+1} = v^k - \lambda \nabla J(v^k) = J_\lambda(v^k)$

- Point fixe de J_λ = point critique de J

Contractivité de J_λ

On suppose que J est α-convexe et que $\nabla J : V \to V$ est Lipschitzienne

$$\exists L > 0, \quad \forall (v,w) \in V \times V, \quad \|\nabla J(v) - \nabla J(w)\|_V \leqslant L\|v - w\|_V$$

Alors, l'application J_λ est contractante lorsque

$$0 < \lambda < \frac{2\alpha}{L^2}$$

- En pratique on ne connait pas ces paramètres ! Guide théorique...

Convergence de la méthode de gradient (2)

- **Preuve :** la contraction vient de l'α-convexité...

$$\begin{aligned}\|J_\lambda(w) - J_\lambda(v)\|_V^2 &= \|(w-v) - \lambda(\nabla J(w) - \nabla J(v))\|_V^2 \\ &= \|w-v\|_V^2 - 2\lambda(\nabla J(w) - \nabla J(v), w-v)_V + \lambda^2\|\nabla J(w) - \nabla J(v)\|_V^2 \\ &\leqslant \rho(\lambda)^2\|w-v\|_V^2\end{aligned}$$

avec $\rho(\lambda)^2 = (1 - 2\lambda\alpha + \lambda^2 L^2)$. Sous la condition $0 < \lambda < 2\alpha/L^2$, on a $0 < \rho < 1$

- Minimum de ρ pour $\lambda_{\rm opt} = \alpha/L^2$, valeur $\rho_{\rm opt} = \left(1 - \dfrac{\alpha^2}{L^2}\right)^{1/2}$

- Soit u le minimiseur global de J sur V (J est α-convexe...). Comme u est point fixe de J_λ, on a

$$u - v^{k+1} = J_\lambda(u) - J_\lambda(v^k)$$

d'où $\|u - v^{k+1}\|_V \leqslant \rho\|u - v^k\|_V$

Convergence de la méthode de gradient (3)

Estimation d'erreur

$$\|u - v^k\|_V \leqslant \rho^k \, \|u - v^0\|_V$$

- L'erreur tend exponentiellement vers zéro...
  - vitesse de convergence dite linéaire : réduire l'erreur d'un ordre de grandeur nécessite un nombre d'itérations constant
  - coût de calcul par itération = essentiellement évaluation de ∇J (méthode d'ordre 1)
  - coût de calcul total = nombre d'itérations x coût par itération

- Lorsque $\alpha \ll L$ (ce qui est souvent le cas en pratique!), $\rho_{\rm opt} \sim 1^-$
$\to$ convergence extrêmement lente

Autres méthodes numériques

- L'efficacité d'un algorithme d'optimisation résulte de
  - sa capacité d'exploration (sortir des puits locaux)
  - sa vitesse de convergence (au sein du bon puits)

- **Algorithme d'ordre zéro** : approches stochastiques
  - exemples : recuit simulé, algorithme génétique
  - bonnes capacités d'exploration, faible vitesse de convergence
  - évaluation de J uniquement, faible coût/itération, bcp. d'itérations

- **Algorithme d'ordre deux** : méthode de Newton (et variantes)
  - faible capacité d'exploration, vitesse de convergence quadratique au voisinage de u : peu d'itérations si bonne initialisation
  - ... mais possibilité de non-convergence "loin" de u
  - évaluation de $\nabla^2 J$ (matrice hessienne), coût/itération élevé

Méthode de gradient : fonctionnelles quadratiques (1)

• Résolution des systèmes linéaires $Au = b$ avec A symétrique définie positive (SDP)

• Minimisation de $J(v) = \frac{1}{2}(v, Av)_{\mathbb{R}^N} - (b, v)_{\mathbb{R}^N}$ dont le gradient est

$$\nabla J(v) = Av - b$$

• Méthode de gradient à pas fixe

$$v^{k+1} = v^k + \lambda d^k, \qquad d^k = b - Av^k =: r^k \text{ (résidu de } v^k\text{)}$$

• Méthode de gradient à pas optimal : optimisation 1D le long de la direction de descente → se souvenir du TD 4 !

• Méthode du gradient conjugué : très efficace pour les systèmes SDP (v^k est minimiseur de J sur un sous-espace affine de dimension k)

Méthode de gradient : fonctionnelles quadratiques (2)

• **Conditionnement** (A matrice SDP)
  - $\kappa(A) \geqslant 1$: rapport entre plus grande et plus petite valeur propre
  - $\kappa(A) \gg 1$: matrice mal conditionnée

• J est α-convexe et ∇J est Lipschitzienne avec
  - α : plus petite valeur propre de A
  - L : plus grande valeur propre de A
  - si A mal conditionnée, alors $\rho_{\mathrm{opt}} \sim 1^-$: convergence très lente

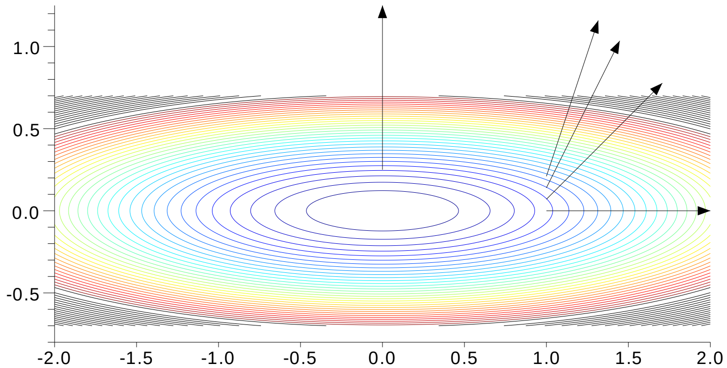
• **Préconditionnement :**
  - matrice SDP P facile à inverser (diagonale, bloc diagonale, ...) avec

$$\kappa(P^{-1/2}AP^{-1/2}) \ll \kappa(A)$$

  - itérer sur le système équivalent (seule P doit être inversée)

$$(P^{-1/2}AP^{-1/2})\tilde{x} = (P^{-1/2}b), \qquad x = P^{-1/2}\tilde{x}$$

Méthode de gradient : fonctionnelles quadratiques (3)



Exemple avec $A = \begin{pmatrix} \varepsilon & 0 \\ 0 & 1 \end{pmatrix}$ et $\varepsilon = 0.07$

Question : que valent $L, \alpha, \kappa(A) =$?

Méthode du gradient projeté

Idée générale

Problème d'optimisation avec contrainte

$$\inf_{v\in K} J(v), \qquad K \subset V \text{ Hilbert}$$

- Algorithme de gradient à pas fixe avec projection à chaque itération

$$v^{k+1} = \Pi_K\Big(v^k - \lambda \nabla J(v^k)\Big)$$

- Projection orthogonale $\Pi_K : V \to K$

$$\|v - \Pi_K(v)\|_V = \inf_{w\in K} \|v - w\|_V$$

- La projection est bien définie si K est convexe et fermé.

$\to$ Pourquoi ?

Projection sur un convexe (1)

- En pratique, il n'est pas simple de calculer la projection (il faut résoudre explicitement le problème de minimisation)...

- Sauf cas particuliers !
  - $K = \overline{B(0,1)}$ auquel cas $\Pi_K(v) =$?
  - $K = [a,b] \subset \mathbb{R}$, auquel cas $\Pi_K(v) =$?

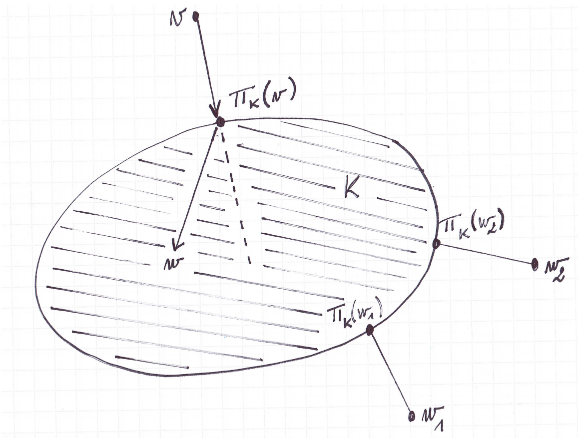
- Pour un ensemble convexe K général, caractérisation de $\Pi_K(v)$?

$\to$ se souvenir de la condition nécessaire

$$\forall w \in K, \qquad (\nabla J_v(u), w-u) \geqslant 0$$

Projection sur un convexe (2)

Remarquer que $\nabla J_v(u) = u - v$ et donc $\langle \Pi_K(v) - v, w - \Pi_K(v)\rangle_V \geqslant 0$



On montre aussi que $\|\Pi_K(v) - \Pi_K(w)\|_V \leqslant \|v - w\|_V$

$\Big[$ Ecrire $\|\Pi_K(v) - \Pi_K(w)\|_V^2 = \underbrace{\langle \Pi_K(v) - v, \dots\rangle_V}_{\leqslant 0} + \langle v - w, \dots\rangle_V + \underbrace{\langle \Pi_K(w) - w, \dots\rangle_V}_{\leqslant 0}\Big]$

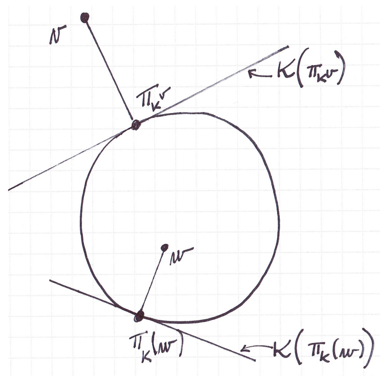
Autre exemple : projection sur une sphère (exercice)

- Problème de minimisation $\inf_K J_v$ où

$$K = \Big\{ s \in V,\ \|s\|_V = 1 \Big\}$$

$$J_v(s) = \frac{1}{2}\|v - s\|_V^2$$

- On note $\Pi_K(v)$ le minimiseur



- Appliquer le résultat précédent pour trouver la projection $\Pi_K(v)$

Algorithme de gradient projeté

- **Initialisation**
  - choisir $v^0 \in K$ (ou $v^0 \in V$ et le projeter)
  - choisir un pas $\lambda > 0$
  - fixer un seuil de convergence $\varepsilon > 0$
- **Itérations**
  - calculer la direction de descente $d^k = -\nabla J(v^k)$
  - appliquer un pas de gradient non-projeté $\widetilde{v}^{k+1} = v^k + \lambda d^k$
  - projeter l'état proposé $v^{k+1} = \Pi_K\left(\widetilde{v}^{k+1}\right)$
  - test de convergence : $\dfrac{\|v^{k+1} - v^k\|_V}{\|v^0\|_V} \leqslant \varepsilon$ ou $\dfrac{|J(v^{k+1}) - J(v^k)|}{J(v^0)} \leqslant \varepsilon$

Convergence vers un point critique

- Reformulation comme un algorithme de point fixe $v^{n+1} = J_{\lambda,K}(v^n)$ avec

$$J_{\lambda,K}(v) = \Pi_K\Big(v - \lambda \nabla J(v)\Big)$$

- Si u est point fixe, alors automatiquement $u = \Pi_K(...) \in K$
- Si K est convexe, u est un point critique de J, cf. propriété projection

$$\Big\langle \Pi_K\left(u - \lambda\nabla J(u)\right) - \left(u - \lambda\nabla J(u)\right), w - \Pi_K\left(u - \lambda\nabla J(u)\right)\Big\rangle_V \geqslant 0$$

soit, pour tout $w \in K$,

$$\langle \nabla J(u), w - u\rangle_V \geqslant 0,$$

qui est la caractérisation d'un point critique sur un ensemble convexe K

Algorithme de gradient projeté : convergence

- Résultat très similaire au gradient simple

Convergence exponentielle de l'erreur si K convexe

- J est α-convexe sur V
- ∇J est Lipschitzienne sur V de constante $L > 0$
- $\lambda \in \left]0, \dfrac{2\alpha}{L^2}\right[$

alors il existe $\rho \in]0,1[$ tel que $\|v^{k+1} - u\|_V \leqslant \rho^k \|v^0 - u^0\|_V$ (u unique minimiseur de $\inf_K J$)

[Preuve : composition des applications contractantes Π_K et $J_\lambda(v) = v - \lambda\nabla J(v)$]

- Intérêt pratique du résultat limité (fonctionnelle pas convexe, $\alpha, L = ?$)

Conclusion

En résumé et en ouverture...

- **Algorithme de gradient :**
  - avec ou sans projection
  - convergence pour J fortement convexe et ∇J Lipschitzienne, si le pas n'est pas trop grand
  - quelques éléments sur la pratique
- **Au menu de la suite :**
  - un TP sur un exemple 2D simple
  - un TD sur la minimisation avec contrainte le 26 mars