Introduction à l'optimisation numérique

Michel TAÏX

Maître de Conférences à l'UPS
LAAS-CNRS, équipe Gepetto

Master AURO

Dans ce cours, nous nous intéressons aux méthodes numériques pour l'optimisation continue, différentiable, linéaire et non linéaire. Après avoir donné les outils mathématiques fondamentaux nous décrirons les différents types de problèmes à résoudre. Pour chacun de ses problèmes nous tenterons de répondre aux questions suivantes :

- Existe-t-il une solution (locale ?) du problème considéré ? si oui, a-t-on unicité ?
- Comment la caractériser ? (conditions d'optimalité).
- Comment la calculer ? Quel type d'algorithme choisir ?

Contenu du cours :

Ce cours comprends trois parties principales :

- Concepts et outils pour l'optimisation
- Optimisation sans contrainte
- Optimisation sous contraintes

Note à l'attention des étudiants

Les transparents ne sont qu'un support de cours, ils comprennent seulement le **minimum** d'information et ne dispensent pas la prise de notes personnelles, bien au contraire. Certaines parties du cours et les exemples seront complétés lors des séances. De nombreuses figures et exemples sont issus de la bibliographie suivante que je conseille à ceux qui veulent aller au delà de ce cours introductif.

- M. Bierlaire. Introduction à l'optimisation différentiable. Presse EPFL.
- S. Boyd, L. Vandenberghe. Convex Optimization. Cambridge Univ. Press.
- R. Fletcher. Practical Methods of optimization, Wiley & Sons.
- C. Lemaréchal. Methodes numeriques d'optimisation, coll. didactique, INRIA ed.
- D. Luenberger. Linear and non-linear optimization, Addison-Wesley.
- M. Minoux. Programmation mathématique. Theorie et algorithmes, Lavoisier.
- J. Nocedal, S. Wright. Numerical Optimization. Springer.

Multiples ressources numériques sur le web, en voici deux à titre d'exemple :
* Convex Optimization, Stephen P. Boyd.
* Numerical Optimization, Moritz Diehl.
Attention à toujours vérifier la source dans ce cas...et ne pas oublier :

L'information n'est pas la connaissance. La seule source de connaissances est l'expérience. Vous avez besoin d'expérience pour acquérir la sagesse (Albert Einstein).

Exemples de logiciels disponibles sur le marché

- CPLEX, XPRESS, Matlab, Gurobi, ... (payant)
- lp_solve, scilab, GLPK, SciPy, qpOASES, OOQP, CASADI... (gratuit)

Introduction à l'optimisation numérique

Part 1 - Concepts et outils de base pour l'optimisation

Problème général

Un problème d'optimisation s'écrit généralement sous la forme :

$$\begin{gathered}\text{Minimiser } f(x) \\ \text{sous la contrainte } x \in S \\ \text{ou bien} \\ \text{sous les contraintes } : g(x) \leq 0 \text{ et } h(x) = 0 \text{ avec} \\ f : \mathbb{R}^n \longrightarrow \Re \\ g : \mathbb{R}^n \longrightarrow \Re^m \\ h : \mathbb{R}^n \longrightarrow \Re^p\end{gathered}$$

Ce problème revient à chercher le (ou les) minimum local (ou global).

Objectif

Étant donnée une fonction $f : \mathbb{R}^n \longrightarrow \Re$ et un ensemble $S \subset \mathbb{R}^n$, trouver $x^* \in S$ tel que $f(x^*) \leq f(x)$ pour tout $x \in S$.

- x^* est appelée un minimum (ou minimiseur) de f sur S.
- Seulement la minimisation car maximiser f revient à minimiser $-f$.
- La fonction f est souvent différentiable (linéaire /non linéaire).
- L'ensemble des contraintes S est défini par un système d'équations et d'inéquation qui peuvent être linéaires ou non linéaires.
- Un point $x \in S$ est dit admissible, acceptable ou réalisable (*feasible*).
- Si $S = \mathbb{R}^n$, alors le problème est dit sans contrainte.

- **Optimisation numérique** : $S \subset \mathbb{R}^n$.
- Optimisation discrète (combinatoire) : S est fini ou dénombrable.
- Commande optimale : S est un ensemble de fonctions (optimisation de trajectoires).
- Optimisation stochastique : données aléatoires.
- Optimisation multicritères : plusieurs fonctions objectifs.

1. 1 Modélisation mathématique

C'est la partie essentielle qui comporte trois étapes :

- Identification des variables de décisions (souvent désignées par un vecteur $x \in \mathbb{R}^n$) : ce sont les paramètres sur lesquels l'utilisateur peut agir pour faire évoluer le système considéré.
- Définition d'une fonction coût (appelée fonction objectif) permettant d'évaluer l'état du système (ex : rendement, performance,...).
- Description des contraintes imposées aux variables de décision.

Définition

Un problème d'optimisation revient à déterminer les variables de décision conduisant aux meilleures conditions de fonctionnement du système (minimiser ou maximiser la fonction coût), tout en respectant les contraintes sur le système.

Objectif	Contraintes	Domaine	Terminologie
Linéaire	Linéaires	Polytope/èdre	Programmation linéaire (P.L.)
Linéaire	Linéaires	$S \subset \mathbb{Z}$	P.L. en nombres entiers
Quadratique	Linéaires	Polytope/èdre	Programmation quadratique
Convexe	Convexes	Convexe	Programmation convexe
Quelconque	Quelconques	Quelconque	Programmation non linéaire (P.N.L.)

Programmation linéaire		
		Méthode du simplexe Algorithme des points intérieurs
Programmation non linéaire		
Sans contrainte	**avec dérivées**	Méthode du gradient Méthodes de Newton et quasi-Newton Gauss-Newton, Levenberg-Marquardt ...
	sans dérivées	Méthodes heuristiques Méthodes stochastiques (recuit simulé,...) ...
Avec contraintes	**avec dérivées**	Gradient projeté Méthode de pénalisation Méthode SQP

1. 2 Applications

L'optimisation intervient dans tous les domaines.

- Contrôle de système, Robotique, Traitement d'image, Reconnaissance des formes, Gestion de production, Finance, ...

Exemples (tableau)

- On dispose d'un ensemble de mesures qui relient la valeur d'une quantité X à celle d'une autre quantité Y. Ces mesures constituent un nuage de points (x_i, y_i) dans le plan (X, Y). On recherche un modèle simple de la relation entre X et Y sous la forme $Y = AX + B$.
- On souhaite construire un hangar parallélépipédique dont le volume est imposé en respectant certaines proportions usuelles et en minimisant le coût de construction. On sait que le hangar doit abriter un volume de $1500m^3$ et que sa largeur doit être égale à 2 fois sa hauteur. Le coût de construction est de N_1 euros le m^2 de mur, N_2 euros le m^2 de plafond, N_3 euros le m^2 de sol.
- On désire minimiser la quantité acheminer de marchandises entre n dépôts à m points de ventes, connaissant :
  * les coûts de transport $c_{ij}(i = 1, \ldots, n; j = 1, \ldots, m)$ entre tous les couples (dépôts, points de vente),
  * les stocks X_i des dépôts, et les niveaux de demande D_j aux points de vente.

1. 3 Rappels et compléments (tableau)

- Dérivation de fonction $f : \Re \longrightarrow \Re$, $y = f(x)$

  $f'(x_0) = \lim_{h \to 0} \frac{f(x_0+h)-f(x_0)}{h}$

  $f(x_0 + h) = f(x_0) + h.f'(x_0) + \frac{h^2}{2^!}.f''(x_0) + \cdots + \frac{h^n}{n^!}.f^{(n)}(x_0) + \mathcal{O}(h^n)$

- Dérivation de fonction $f : \mathbb{R}^n \longrightarrow \Re$, $y = f(\underline{x})$

  $f(\underline{x}) = f(\underline{x}_0) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(\underline{x}_0).(x_i - (x_0)_i) \Longrightarrow$ définit plan tangent.

- Gradient de f : $\nabla f(\underline{x}) = (\frac{\partial f}{\partial x_1}(\underline{x}), \frac{\partial f}{\partial x_2)}(\underline{x}), ..., \frac{\partial f}{\partial x_n}(\underline{x}))^t$

  $f(\underline{x}) = f(\underline{x}_0) + [\frac{\partial f}{\partial x}(\underline{x}_0)]^t.(\underline{x} - \underline{x}_0) + \frac{1}{2}(\underline{x} - \underline{x}_0)^t.\frac{\partial^2 f}{\partial x.\partial x^t}(\underline{x}_0).(\underline{x} - \underline{x}_0)$

  $f(\underline{x}) = f(\underline{x}_0) + (\underline{x} - \underline{x}_0)^t.\nabla f(\underline{x}) + \frac{1}{2}(\underline{x} - \underline{x}_0)^t.\nabla^2 f(\underline{x}).(\underline{x} - \underline{x}_0)$

- Matrice Hessienne de f (symétrique) :

$$\nabla^2 f(\underline{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Dérivée directionnelle

Dérivée directionnelle de f en x dans la direction d :
$f : \mathbb{R}^n \longrightarrow \Re$ avec $x, d \in \mathbb{R}^n$.
$D_d f(x) = \lim_{h \to 0, h \neq 0} \frac{f(x+hd)-f(x)}{h}$

Direction de descente

La direction d est une direction de descente si $d^t.\nabla f(x) < 0$

Theorem

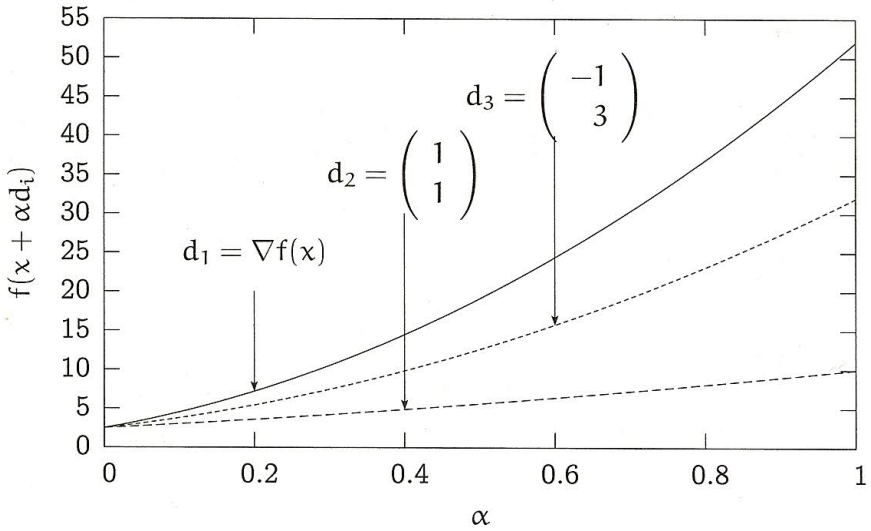
Soit $f : \mathbb{R}^n \longrightarrow \Re$ différentiable. Soient $x, d \in \mathbb{R}^n$ avec $\nabla f(x) \neq 0$. Si d est une direction de descente alors il existe $\eta > 0$ tel que :

$$f(x + \alpha.d) < f(x), \forall 0 < \alpha < \eta$$

Soit $d^* = -\nabla f(x)$ alors $\forall d \in \mathcal{R}^n \mid \|d\| = \|\nabla f(x)\|$:
$d^*.\nabla f(x) \leq d^t.\nabla f(x)$

Exemple

Soit la fonction $f(x) = \frac{1}{2}.x_1^2 + 2.x_2^2$ et le point $x = (1,1)^t$



Fonctions quadratiques

Une fonction $f : \mathbb{R}^n \longrightarrow \Re$ est dite quadratique si elle peut s'écrire $f(\underline{x}) = \frac{1}{2}.\underline{x}^t.Q.\underline{x} + G^t.\underline{x} + c$ avec Q matrice symétrique $n \times n$, $G \in \mathbb{R}^n$ et $c \in \Re$.

Dans ce cas : $\nabla f(\underline{x}) = Q\underline{x} + G$ et $\nabla^2 f(\underline{x}) = Q$

1. 4 Vitesse de convergence d'un algorithme

- Algorithme : $F : U \longrightarrow U$ tel que : $u_{k+1} = F(u_k)$
- Point fixe : $F^{\infty}(u) = \{u \in U \mid F(u) = u\}$
- Point fixe attractif
- Un algorithme converge si ses points fixes sont des candidats à la solution du problème et si leur bassin d'attraction recouvre la totalité de U.
- La vitesse de convergence mesure la décroissance vers 0 de la distance entre les valeurs engendrées et leur limite.
- Convergence linéaire : $\exists \alpha \in [0,1], k_1 \geq 1, \mid \forall k \geq k_1 \quad \frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|} \leq \alpha$
- Convergence superlinéaire : $\exists k_1 \geq 1, \mid \forall k \geq k_1 \quad \frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|} \longrightarrow 0$
- Convergence superlinéaire d'ordre p :
  $\exists M > 0, k_1 \geq 1, \mid \forall k \geq k_1 \quad \frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|^p} \leq M$

1. 5 Minimum

Minimum local/global

- $x^* \in S$ est un minimum global si $f(x^*) \leq f(x)$ pour tout $x \in S$.
- $x^* \in S$ est un minimum local si $f(x*) \leq f(x)$ pour $x \in B(x^*, \epsilon)$ (voisinage de x^*). On a un minimum local strict si $<$.

Exemples : unicité/infinité de minima local/global (tableau)

- Trouver (ou même vérifier) un minimum global est en général très difficile.
- La plupart des méthodes d'optimisation sont conçues pour trouver un minimum local (qui peut-être ou non global).
- Si un minimum global est recherché, on peut essayer d'appliquer une méthode d'optimisation avec des points initiaux différents.
- Pour certains problèmes tels que la programmation linéaire, la recherche d'un minimum global est atteignable.

Existence d'un minimum

- Si f est continue sur un fermé borné $S \subset \mathbb{R}^n$ alors f admet un minimum global sur S.
- Si S n'est pas fermé ou non borné, alors f peut n'avoir ni minimum local, ni minimum global sur S.
- Une fonction continue f sur un ensemble non borné S est dite coercive si $\lim_{\|x\| to \infty} f(x) = +\infty$
  i.e. $f(x)$ devient grand quand $\|x\|$ devient grande
- Si f est coercive sur un ensemble fermé non borné $\subset \mathbb{R}^n$ alors f admet un minimum global sur S

1. 6 Convexité

- Ensemble convexe : l'ensemble S est convexe s'il contient tous les segments compris entre deux de ses points.
  $\forall x, y \in S, \forall \alpha \in [0,1]$ alors $\alpha x + (1-\alpha)y \in S$
- Une fonction $f : S \subset \mathbb{R}^n \longrightarrow S$ est convexe sur l'ensemble convexe S si $\forall x, y \in S$, $\forall \alpha \in [0,1]$ alors
  $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) \in S$
- Tout minimum local d'une fonction convexe f sur un ensemble convexe $S \subset \mathbb{R}^n$ est un minimum global de f sur S.
- Tout minimum local d'une fonction strictement convexe f sur un ensemble convexe $S \subset \mathbb{R}^n$ est le minimum global (unique) de f sur S.

Soit $f : X \subseteq \mathbb{R}^n \longrightarrow \Re$ de classe C^2 avec X ouvert convexe.
Si $\nabla^2 f(\underline{x})$ est définie positive (semi-definie) pour tout x dans X alors f est strict. convexe (convexe)

1. 7 Optimisation d'une fonction d'une variable réelle

Souvent on ne dispose que d'une informations locale d'une fonction et pas de représentation globale, supposons cette fonction unimodale.
Une fonction f réelle à valeur dans $I = [a, b]$ est dite unimodale sur I si :
* elle admet un minimum unique x^* dans I
* elle est strict. décroissante sur $[a, x^*[$ et strct. croissante sur $]x^*, b]$

Méthode par dichotomie

- Décomposer $[a, b]$ en 4 intervalles $a = x_1 < x_2 < x_3 < x_4 < x_5 = b$
- Calculer f aux points x_i
- Traiter les 5 cas possibles pour réduire de moitié $[a, b] = [x_1, x_5]$
  - $f(x_1) < f(x_2) < f(x_3) < f(x_4) < f(x_5)$
  - $f(x_1) > f(x_2) < f(x_3) < f(x_4) < f(x_5)$
  - $f(x_1) > f(x_2) > f(x_3) < f(x_4) < f(x_5)$
  - $f(x_1) > f(x_2) > f(x_3) > f(x_4) < f(x_5)$
  - $f(x_1) > f(x_2) > f(x_3) > f(x_4) > f(x_5)$
- Continuer sur le nouveau intervalle

Méthode de la section dorée

f unimodale sur $[a, b]$ et soient x_1, x_2 deux points de $[a, b]$ avec $x_1 < x_2$. En comparant $f(x_1)$ et $f(x_2)$, on peut retirer un des intervalles $]x_2, b]$ ou $[a, x_1[$, On veut réduire l'intervalle de recherche d'un même facteur à chaque itération et de plus, on veut conserver les mêmes relations entre les points du nouvel intervalle qu'avec celui de l'ancien. Pour se faire, on choisit les positions relatives des deux points par r et $1 - r$ avec $r^2 = 1 - r$. Donc $r = \frac{(\sqrt{(5)}-1)}{2}$ $(\simeq 0.618)$ A chaque itération, on n'effectue qu'une seule évaluation de la fonction La taux de convergence est linéaire

---

```
1  x1 = a + (1 − r)(b − a); f1 = f(x1) ; x2 = a + r.(b − a); f2 = f(x2) ;
2  while tol do
3  |   read ;
4  |   if f1 > f2 then
5  |   |   a = x1, x1 = x2, f1 = f2 ;
6  |   |   x2 = a + r.(b − a); f2 = f(x2) ;
7  |   else
8  |   |   b = x2, x2 = x1, f2 = f1 ;
9  |   |   x1 = a + (1 − r)(b − a); f1 = f(x1) ;
10 |   end
11 end
```

L'optimisation numérique non linéaire peut se décomposer en deux parties :

1. Etude des conditions d'optimalité
2. Méthode de recherche

Conditions d'optimalité

- Analyse théorique du problème
- Source d'inspiration des algorithmes de résolution
- Fournissent des éléments pour déterminer un critère d'arrêt des algorithmes

Méthodes de recherche

- Méthodes numériques itérative : $x_{k+1} = x_k + \alpha.d$
- la recherche linéaire
- la région de confiance

1. 8 Méthodes de résolution

De manière générale, on ne dispose pas de moyen formel permettant de détecter tous les points critiques.
Il faut alors mettre en œuvre des méthodes numériques itératives dont l'objectif est de converger vers un extremum local en utilisant les conditions d'optimalité. Elles sont basées sur l'évaluation de la fonction coût et de ses dérivées (premier ordre et parfois au second), à partir de la donnée d'un jeu de paramètres initiaux.
Les deux principales stratégies pour trouver un minimum sont :

- la recherche linéaire (*line search*) : l'algorithme choisit une direction à chaque itération k, d_k, pour chercher dans cette direction la nouvelle valeur de x_k qui minimise f. Il faut donc déterminer le pas α_k le long de la direction d_k.
- la région de confiance (*trust region*) : on utilise f pour construire un modèle de la fonction, m_k, qui se comporte "comme" la fonction au voisinage de x_k. Ce modèle n'est valide que dans une région proche de x_k, région de confiance. Connaissant la région, c.a.d la distance maximale de déplacement, on cherche la direction de descente.

Pourrez-vous sauver le monde ?

Vous devez désamorcer une bombe nucléaire sur un bateau amarré à 50 m du rivage. Vous vous trouvez à 100 m du point le plus proche du bateau sur la plage. Votre vitesse de course sur la plage est de 18 km/h et votre vitesse de nage de 10 km/h. Sachant qu'il faut appuyer sur un bouton pour désamorcer la bombe et que celle-ci est programmée pour exploser dans 35 secondes, aurez-vous le temps de sauver le monde ? (exemple inspiré de Walker)

Quelles sont les variables de décision ?
Comment résoudre le problème sous forme d'un problème d'optimisation ?
Avez-vous des contraintes ?

Part 2 - Optimisation sans contrainte
A/ Conditions

Soit (P_0) le problème d'optimisation suivant

$$\min \ f(x)$$

$$x \in \mathbb{R}^n$$

où f est une fonction deux fois différentiable.

2. 1 Conditions d'optimalité sans contrainte

Pour des fonctions d'une variable, on trouve les extrema en calculant les zéros de la dérivée.

Pour des fonctions à n variables, on cherche les points critiques, i.e. les solutions du système : $\nabla f(\underline{x}) = 0$ avec $\nabla f(\underline{x})$ qui est le gradient de f.

Théorème CN1 (condition nécessaire d'optimalité du 1^{er} ordre)

Soit x^* un minimum local du problème (P_0) alors

$$\nabla f(x^*) = 0$$

⚠

Sketch of proof

Pas de direction de descente qui améliore f (dont $\nabla f(x^*)$).

Théorème CN2 (condition nécessaire d'optimalité du 2^{er} ordre)

Soit x^* un minimum local du problème (P_0) alors

$$\nabla^2 f(x^*) \geq 0$$

⚠

Sketch of proof

Développement de Taylor

$$\begin{array}{rcl} f(x^* + \alpha d) - f(x^*) & = & \alpha d^t \nabla f(x^*) + \frac{1}{2}\alpha^2 d^t \nabla^2 f(x^*) d + o(\|\alpha d\|^2) \\ & = & \frac{1}{2}\alpha^2 d^t \nabla^2 f(x^*) d + o(\|\alpha d\|^2) \\ & \geq & 0 \end{array}$$

$\Rightarrow \nabla^2 f(x^*)$ est semi-définie positive.

Condition nécessaire du second ordre : cas sans contrainte

Pour une fonction $f : S \subset \mathbb{R}^n \longrightarrow S$ de classe C^2, on distingue les points critiques en considérant la matrice Hessienne $H_f(x)$ définie par

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix} \quad (H_f(x)_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j})$$

Cette matrice est symétrique.

A un point critique x, si $H_f(x)$ est :

- définie-positive alors x est un minimum de f (val propre >0, ou $\forall x \neq 0, x^t H x > 0$)
- définie-négative alors x est un maximum de f
- indéfinie alors x est un point selle

Théorème (condition suffisante d'optimalité locale)

Soit x^* un point vérifiant les conditions suivantes

$$\nabla f(x^*) = 0$$

$$\nabla^2 f(x^*) > 0.$$

Alors x^* est un minimum local du problème (P_0).

Sketch of proof

Développement de Taylor.

Les éléments x qui respectent CN1 sont appelés **points critiques** ou points stationnaires. Parmi eux se trouvent des minima locaux, des maxima locaux et des points qui ne sont ni l'un ni l'autre. Ces derniers sont appelés des points de selle.

En pratique, la CN2 est difficile à vérifier systématiquement car elle nécessite de calculer les dérivées seconde et d'analyser les valeurs propres de la matrice hessienne.

Les résultats énoncés précédemment concernent uniquement l'optimalité locale. Il n'y a pas de résultat sur l'optimalité globale en dehors des fonctions convexes.

Théorème (condition suffisante d'optimalité globale)

Soit x^* un minimum local de (P_0). Si f est continue et convexe alors x^* est un minimum globale.
Si de plus f est strictement convexe, x^* est unique.

Conditions d'optimalité pour les problèmes quadratiques

Soit (P) le problème d'optimisation suivant

$$\min \ f(x) = \frac{1}{2}.x^t.Q.x + g^t.x + c$$

avec Q matrice symétrique $\in \mathbb{R}^{n\times n}$, $g \in \mathbb{R}^n$ et $c \in \mathbb{R}$

1. Si Q n'est pas semi-définie positive, alors le problème ne possède pas de solution. Il n'existe aucun $x \in \mathbb{R}^n$ qui soit un minimum local de f
2. Si Q est définie positive alors :
   $x^* = -Q^{-1}.g$ est l'unique minimum global.

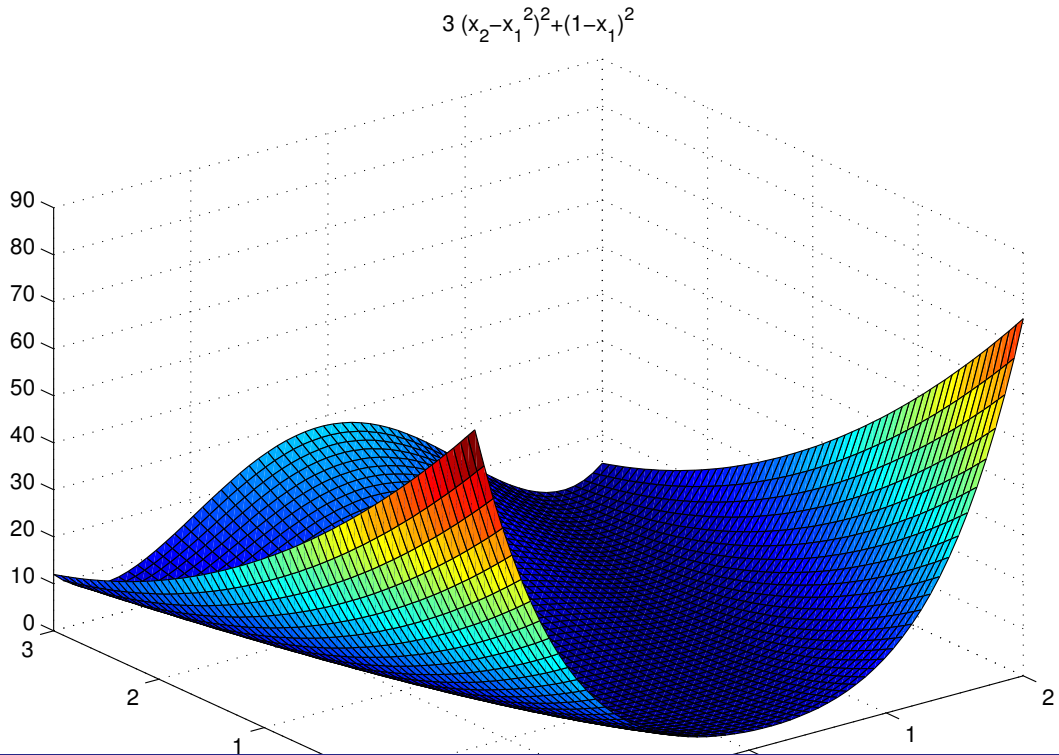
Exemple 1

$$f(x_1, x_2) = x_1^2 - x_2^2$$

Exemple 2

$$f(x_1, x_2) = 3(x_2 - x_1^2)^2 + (1 - x_1)^2$$

en $(1, 1)$.



Exemple 3

$$f(x_1, x_2) = -x_1^4 - x_2^4$$

en $(0, 0)$.

Exemple 4

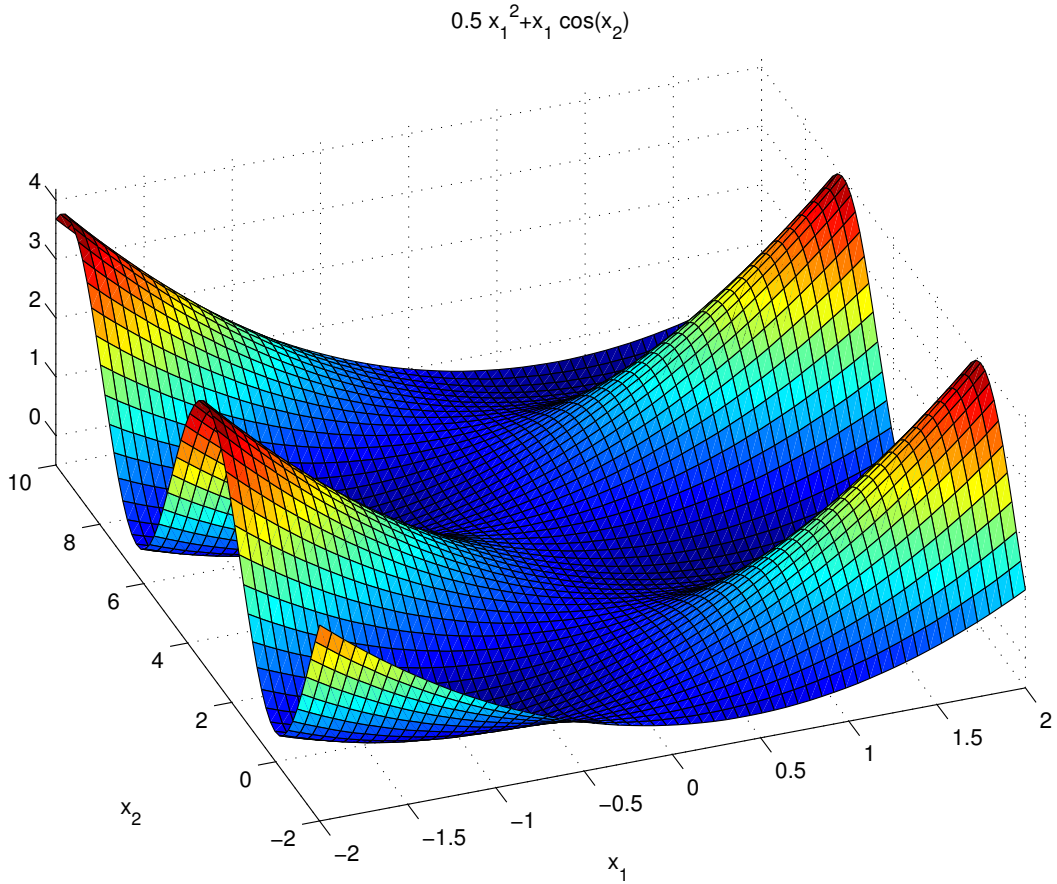
$$f(x_1, x_2) = \frac{1}{2}x_1^2 + x_1 cos x_2$$

en $(-1, 0)$, $(1, \pi)$, $(0, \pi/2)$ puis $(-1, 2\pi)$.

Exemple 4

$$f(x_1, x_2) = \frac{1}{2}x_1^2 + x_1 cos x_2$$

en $(-1, 0)$, $(1, \pi)$, $(0, \pi/2)$ puis $(-1, 2\pi)$.



Part II - Optimisation sans contrainte
B/ Méthodes

Nous allons maintenant étudier les méthodes de recherche pour les problèmes non contraints.

Problème :

$$\min_{x \in \mathbb{R}^n} f(x)$$

Solution : Partir d'un point x_0 et construire une suite $x_1, x_2, \ldots x_n$ qui converge vers un optimum local x^*.

Méthode de recherche linéaire

$$x_{k+1} = x_k + \alpha d$$

d est une direction

α est un pas

Définition (direction de descente)

d est une direction de descente, s'il existe $\alpha > 0$ tel que

$$f(x_{k+1}) < f(x_k)$$

Algorithme général

Repeter

$d_k \leftarrow$trouver une direction

$\alpha \leftarrow$ calculer un pas

$x_{k+1} \leftarrow x_k + \alpha d_k$

$k = k + 1$

tant que (condition d'arrêt = faux)

La condition d'arrêt peut prendre plusieurs formes :

- $\|\nabla f\| \leq \epsilon$
- $\|f(x_{k+1}) - f(x_k))\| \leq \epsilon$
- $\|x_{k+1} - x_k\| \leq \epsilon$
- $k = K_{max}$

En pratique, il est préférable de travailler avec des erreurs relatives plutôt qu'avec des erreurs absolues, trop dépendantes de l'échelle des valeurs.

Théorème (descente de gradient)

L'opposé du gradient est une direction de descente : $d = -\nabla f(x)$

Preuve

Approximation linéaire de $f(x_k + \alpha d_k)$.

$$f(x_k + \alpha d) = f(x_k) + \alpha d_k^t \nabla f(x_k) + o(\alpha)$$

si $d_k = -\nabla f(x_k)$, alors $f(x_{k+1}) = f(x_k) - \alpha \|\nabla f(x_k)\|^2 + o(\alpha)$

La méthode du gradient

La pente est donnée par l'opposé du gradient

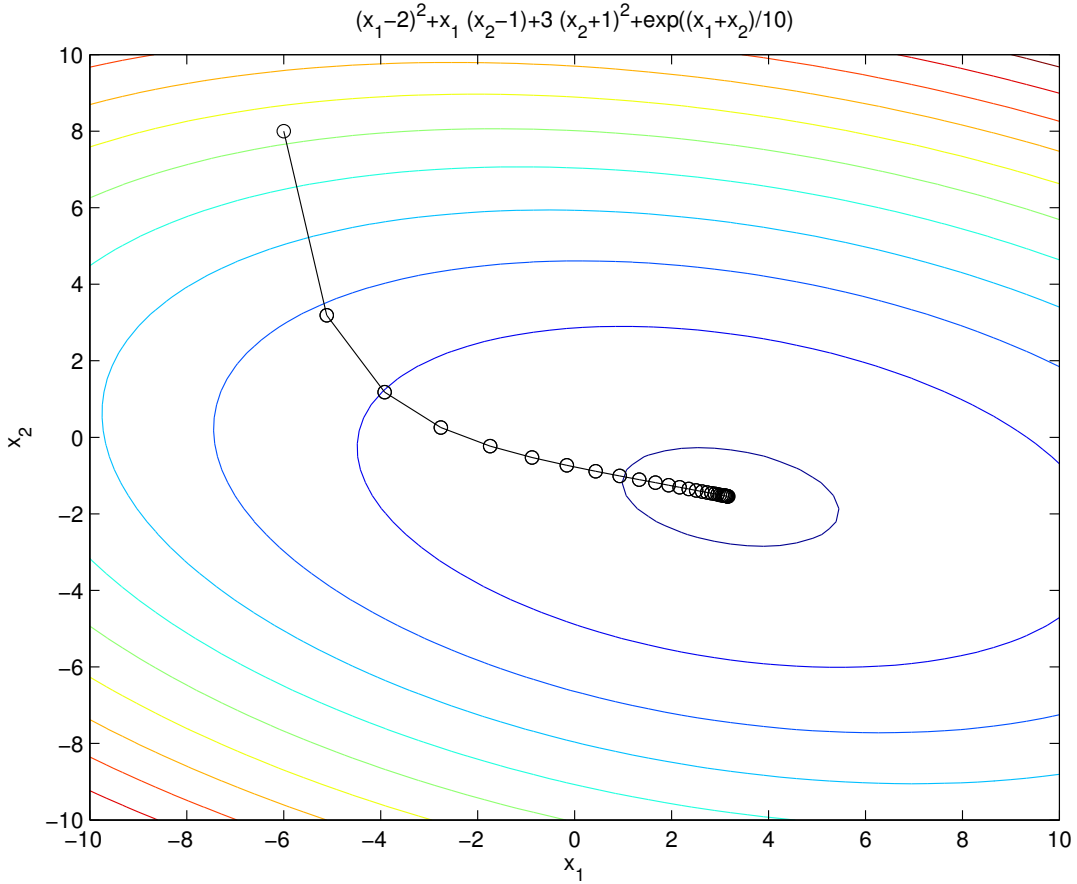
$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

Quelle valeur pour le pas α ?

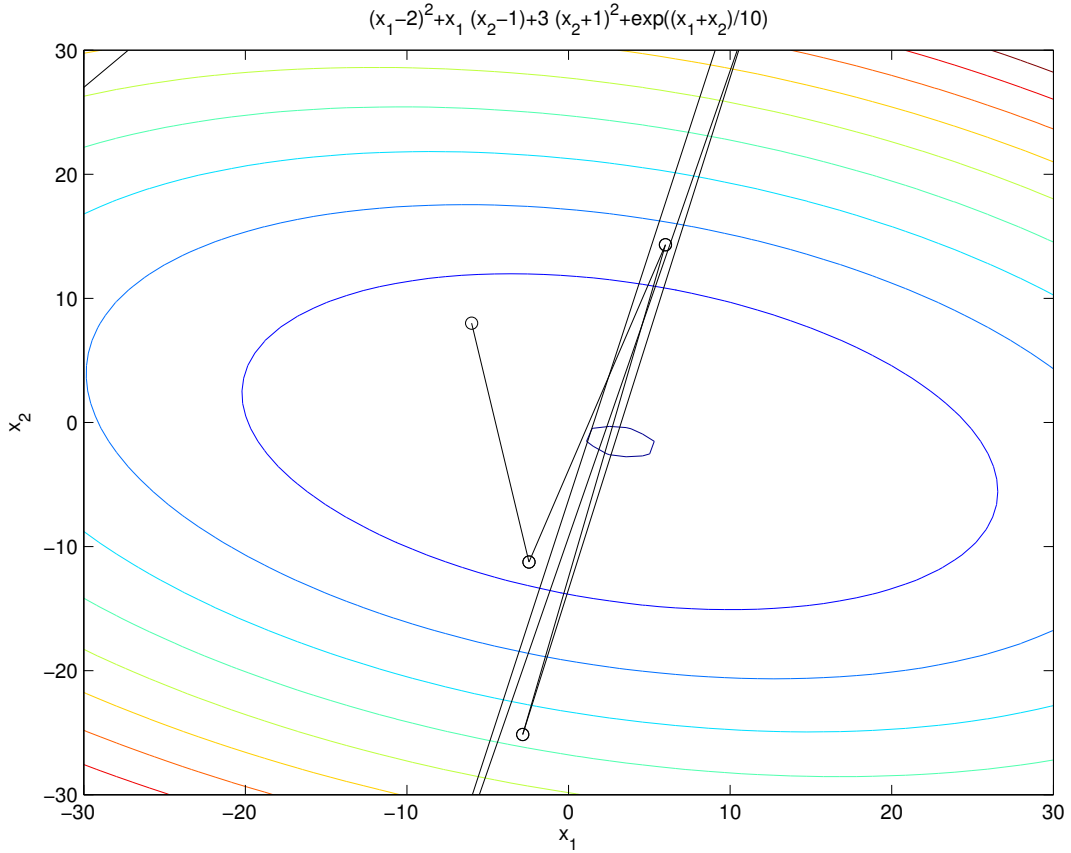
1. α constant

$\rightarrow$ mauvaise idée...

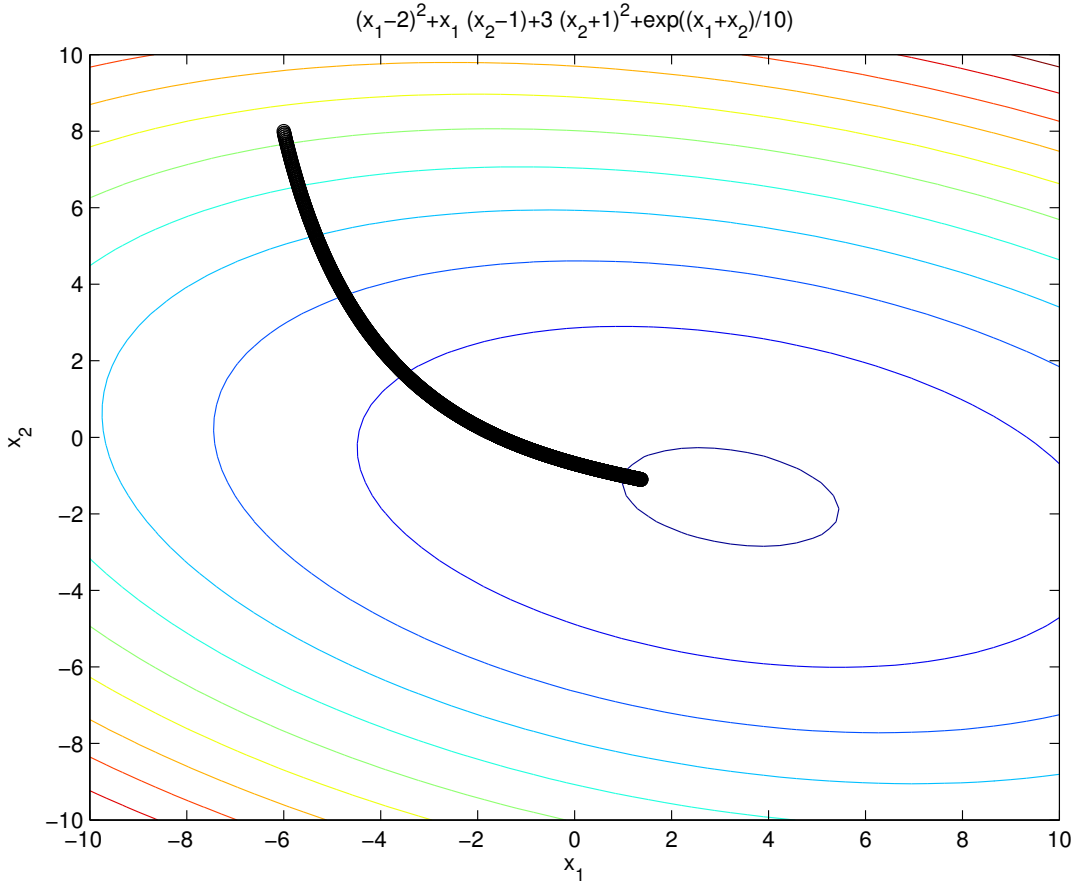
α bien calibré (28 itérations, $\|f(x_{k+1}) - f(x_k))\| \leq 0.001$)



α trop grand (10 itérations, $\|f(x_{k+1}) - f(x_k))\| = NaN$)



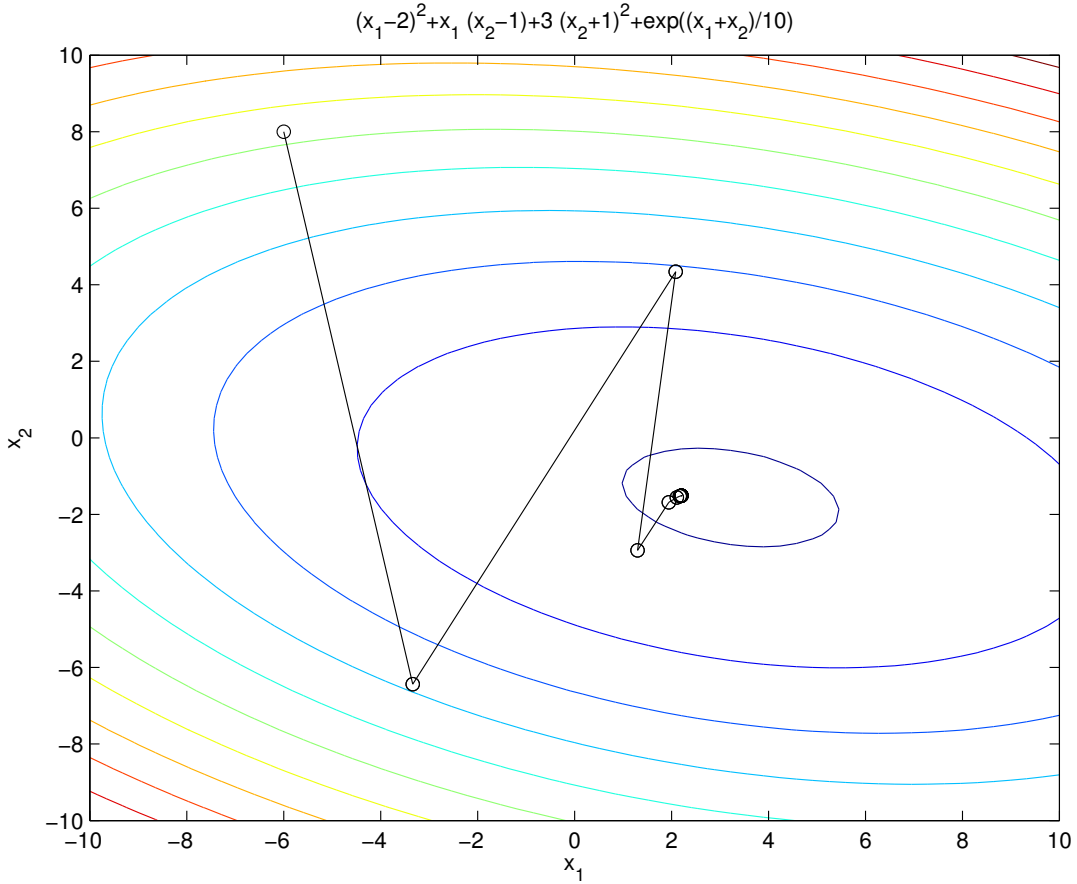
α trop petit (1000 itérations, $\|f(x_{k+1}) - f(x_k))\| = 0.0113$)



2. α variable

- $\alpha = \frac{\alpha_0}{k+1}$
- $\alpha = \frac{\alpha_0}{\sqrt{k+1}}$

peu performant, difficile à faire converger.



3. α variable optimal

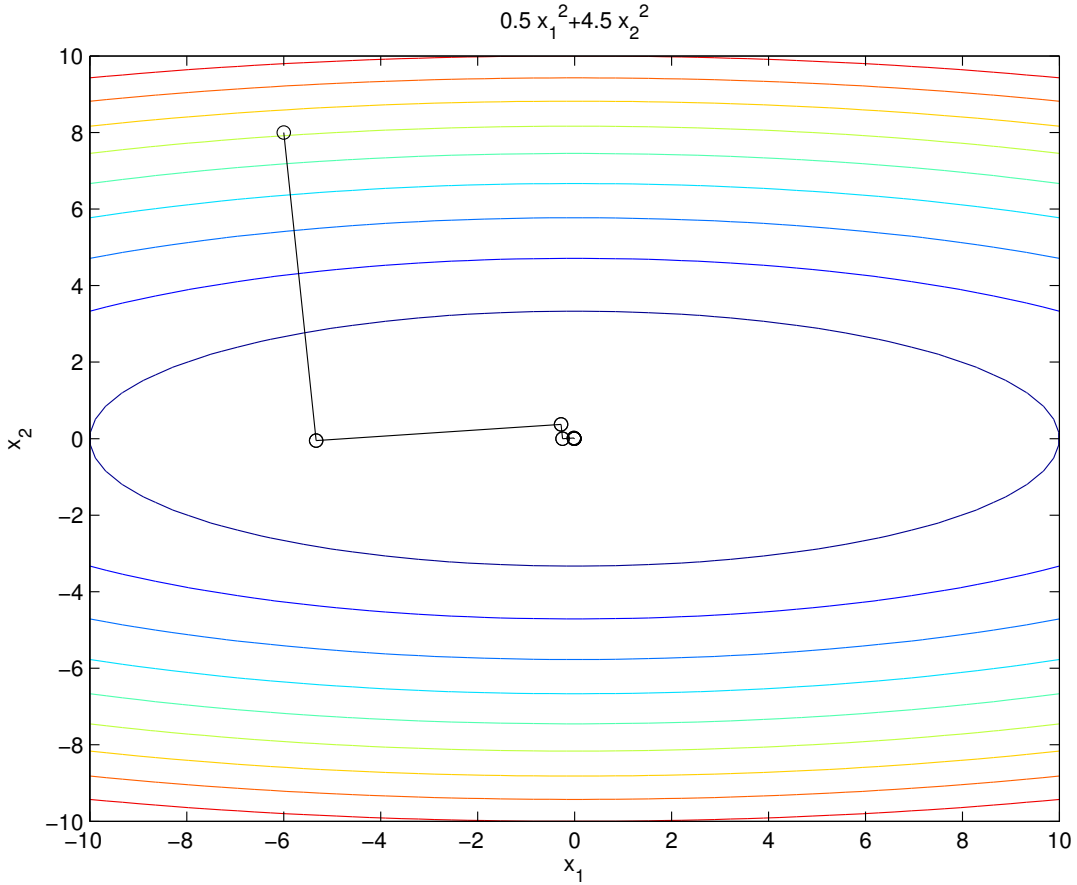
$$arg \min_{\alpha \geq 0} f(x_k - \alpha \nabla f(x_k)) \tag{1}$$

Méthode plus connue sous le nom de *plus forte pente* ("steepest descent") ou *gradient à pas optimal*. Le problème (1) est un problème d'optimisation unidimensionnel.

Calculer le pas optimal qui minimise la fonction

$$f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2.$$

7 itérations, $\|f(x_{k+1}) - f(x_k))\| \leq 10^{-5}$.



- simple, robuste mais lent
- résolution du problème (1) parfois coûteux bien qu'il n'y ait qu'une seule variable.
- Si la direction n'est pas bonne, il n'est pas judicieux de dépenser beaucoup d'effort pour résoudre le problème.
- Convergence en zig-zag

Conditionnement

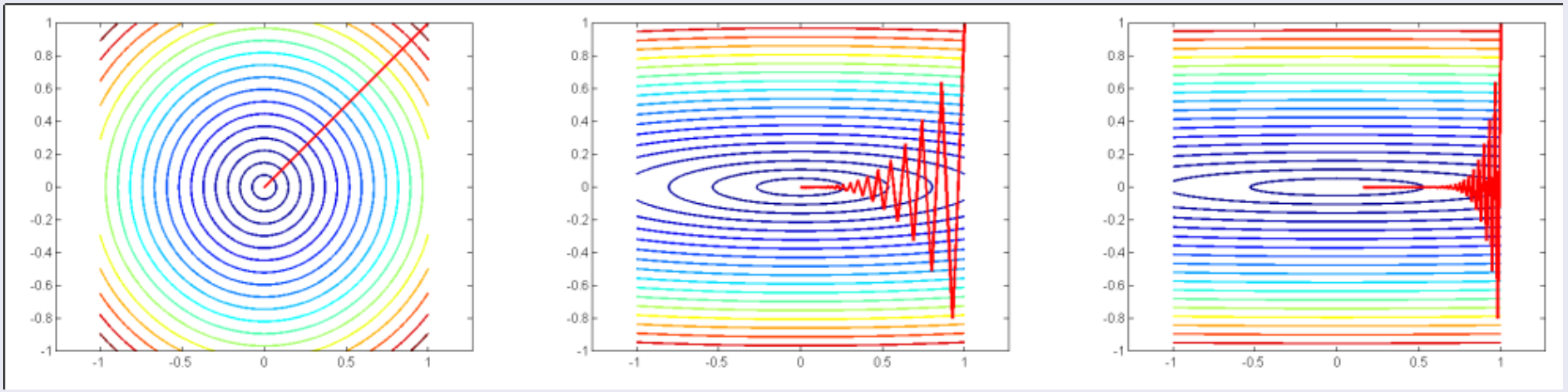
Soit $f : \mathbb{R}^n \to \mathbb{R}$ fonction deux fois différentiable et soit $x \in \mathbb{R}^n$. Le conditionnement de f en x est le nombre de conditionnement κ de $\nabla^2 f(x)$

Si A matrice inversible alors $\kappa(A) = \|A^{-1}\|.\|A\|$
$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$ (rapport entre la plus grande/plus petite valeur singulière)
Si A est normale alors $\kappa(A) = \frac{|\lambda_{max}(A)|}{|\lambda_{min}(A)|}$

(A normale $\iff \exists$ matrice U unitaire $\mid U^{-1}AU$ soit diagonale)

Soit $f : \mathcal{R}^n \to \mathcal{R}$ fonction deux fois différentiable et soit $x \in \mathbb{R}^n$.
Soit le changement de variable $x' = M.x$ et une fonction f' telle que $f'(x') = f(M^{-1}.x')$
Préconditionner f en x revient à définir M telle que le conditionnement de f' en Mx soit meilleur que celui de f en x.

Conditionnement



Exemple : tableau $f(x) = \frac{1}{2}x^t.Q_\alpha.x$ avec $Q_\alpha = \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix}$ et $\alpha \geq 1$

Conditions sur la longueur de pas

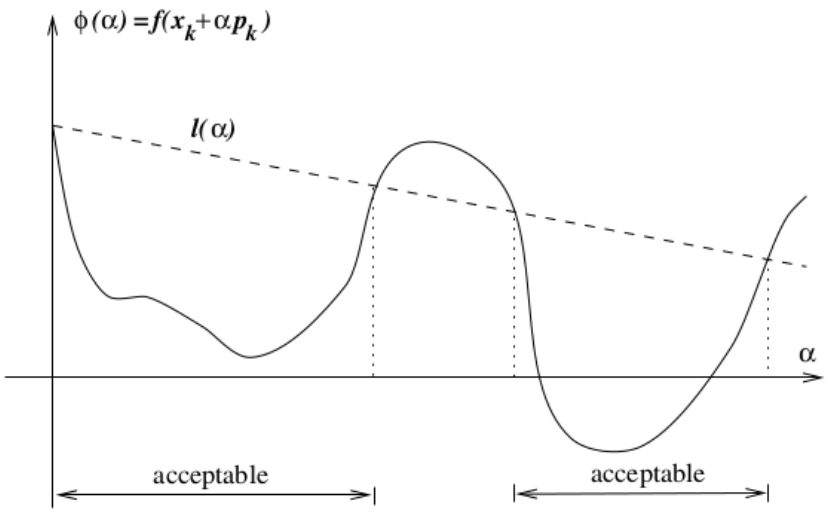
Petit/grand pas : approximation linéaire / convergence rapide
Relation entre la longueur du pas et la diminution de la fonction objectif
Idée : diminution de la fonction objectif proportionnelle à la taille du pas

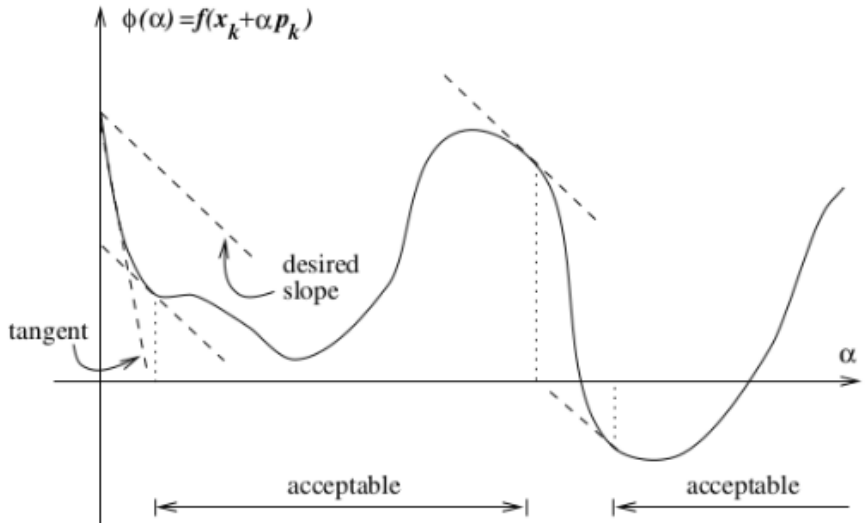
Première condition de Wolfe -Armijo (pas trop grand ?)

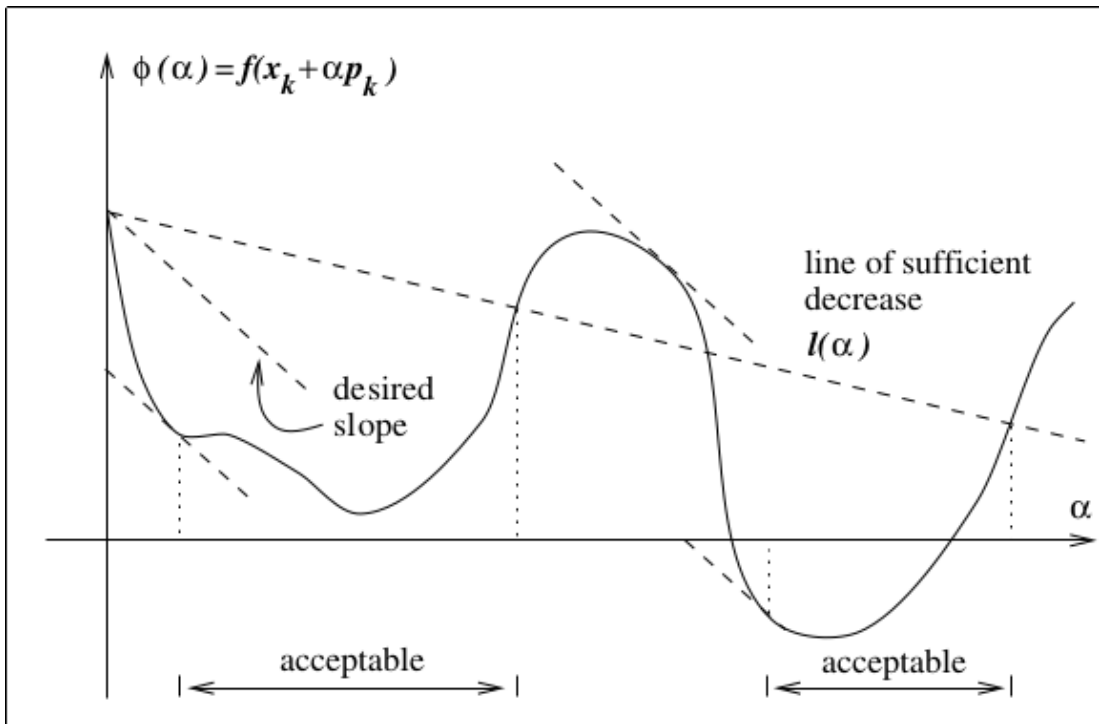
Soient $f : \mathcal{R}^n \to \mathcal{R}$ fonction différentiable, $x_k \in \mathbb{R}^n$, $d_k \in \mathbb{R}^n$ une direction de descente telle que $\nabla f(x_k)^t.d_k < 0$ et un pas $\alpha_k > 0$.
f diminue suffisamment en $x_k + \alpha_k.d_k$ par rapport à x_k si :
$f(x_k + \alpha_k.d_k) \leq f(x_k) + \alpha_k.\beta_1.\nabla f(x_k)^t.d_k$ avec $0 < \beta_1 < 1$.

Seconde condition de Wolfe (pas trop petit ?)

Soient $f : \mathcal{R}^n \to \mathcal{R}$ fonction différentiable, $x_k \in \mathbb{R}^n$, $d_k \in \mathbb{R}^n$ une direction de descente telle que $\nabla f(x_k)^t.d_k < 0$ et un pas $\alpha_k > 0$.
Le point $x_k + \alpha d_k$ apporte un progrès par rapport à x_k si :
$\nabla f(x_k + \alpha d_k)^t.d_k \geq \beta_2.\nabla f(x_k)^t.d_k$ avec $0 < \beta_2 < 1$.







(Nocedal & Wright)

Soit $\cos(\theta_k)$ l'angle entre la direction du gradient/direction de descente d_k : $\cos(\theta_k) = \frac{-\nabla f(x_k)^t . d_k}{\|\nabla f(x_k)\| . \|d_k\|}$

Convergence globale des algorithmes de descente et pas de Wolfe

Soit $f : \mathbb{R}^n \to \mathbb{R}$ fonction différentiable, de gradient Lipschitz ($\|\nabla f(x) - \nabla f(y)\| \leq L.\|x - y\|$). On suppose que f est borné inférieurement dans $\mathbb{R}^n$.
Soit un algorithme de descente définis par : $x_{k+1} = x_k + \alpha_k d_k$ avec d_k la direction de descente et $\alpha_k > 0$ le pas vérifiant les conditions de Wolfe.

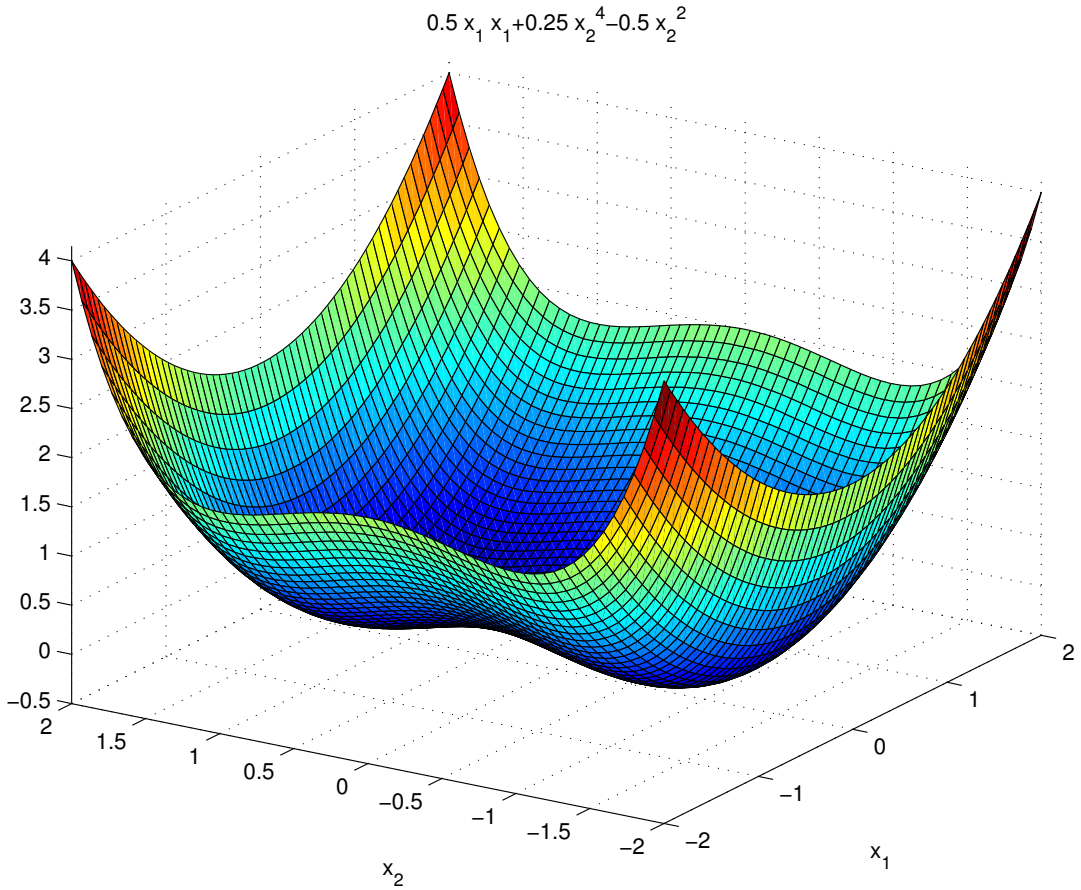
alors $\sum \cos^2(\theta_k).\|\nabla f(x_k)\|^2$ converge. (Théorème de Zoutendijk)

Donc si la direction du gradient $\nabla f(x_k)^t$ n'est pas orthogonale à la direction de descente d_k : $\exists a > 0, \forall k \in \mathbb{N}, \cos(\theta_k) \geq a$
Alors la série $\sum \|\nabla f(x_k)\|^2$ converge.

Ceci permet de montrer que sous certaines conditions l'algorithme du gradient converge globalement (besoin de connaître L !)

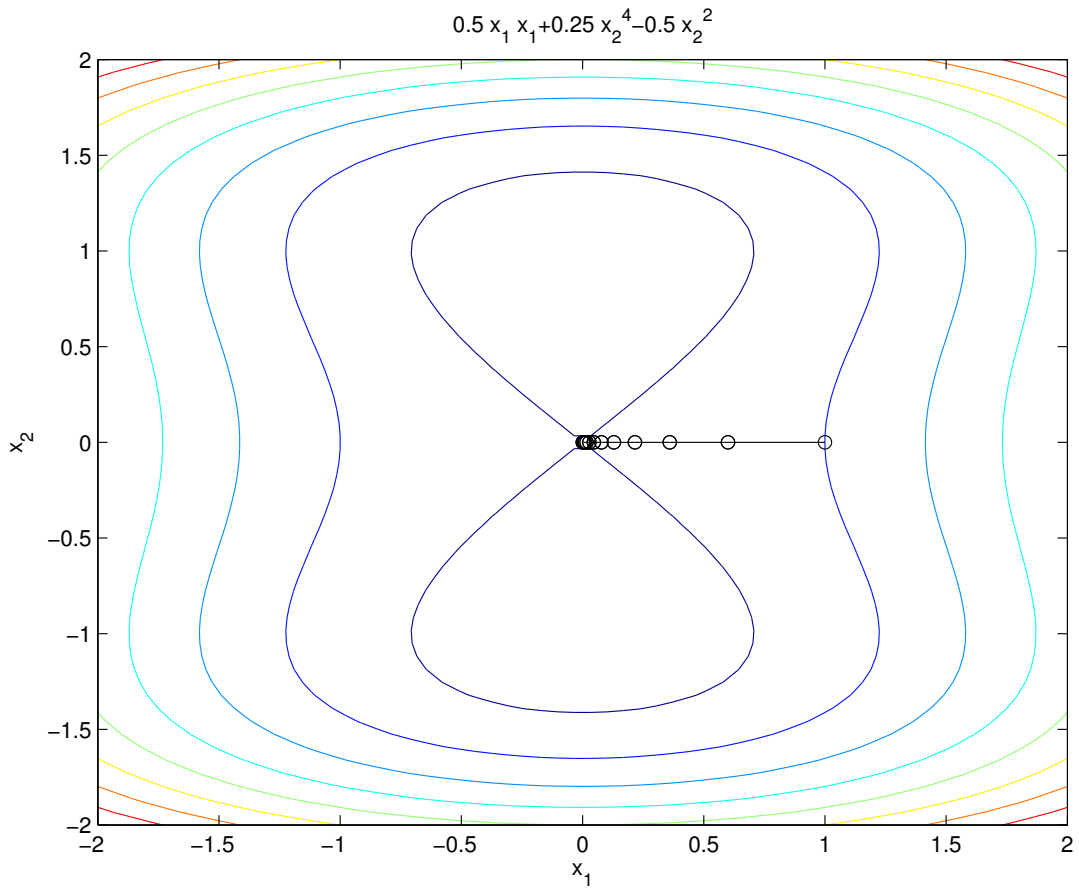
La méthode du gradient est une méthode du premier ordre : elle vise à annuler le gradient... mais n'atteint pas nécessairement un optimum local !

$$f(x) = \frac{1}{2}x_1^2 + \frac{1}{4}x_2^4 - \frac{1}{2}x_2^2$$



Départ en $x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

Présence de deux optimum locaux cependant pas de convergence vers un optimum local.



Problèmes quadratiques et gradient conjugué

Soit $\min f(x) = \frac{1}{2}x^t Qx + b^t x + c$ avec $Q > 0$ (définie positive).
L'unique minimum de la fonction est donnée par le système d'équation suivant :

$$Qx = -b \Longrightarrow \text{calcul de } Q^-1$$

Méthode directe

Résoudre ce système (sans inverser Q, trop couteux) en utilisant la factorisation de Cholesky.

Soit $Q \in \mathbb{R}^{nxn}$ une matrice symétrique définie positive.
La décomposition de Cholesky de Q est : $Q = L.L^t$
avec $L \in \mathbb{R}^{nxn}$ une matrice triangulaire inférieure.

On montre que $Q^{-1} = (L^{-1})^t.(L^{-1})$
(calcul d'inverse de matrice triangulaire efficace)

La méthode directe reste cependant coûteuse. La méthode du gradient conjugué est une alternative moins gourmande.

Méthode des directions conjuguées

Soit $Q \in \mathbb{R}^{n \times n}$, définie positive.
Les vecteurs non nuls $d_1, \ldots, d_k$ sont Q-conjuguées si

$$d_i^t Q d_j = 0, \qquad \forall i, j \text{ tels que} i \neq j.$$

Corollaire

Si $Q = Id$, les directions conjuguées sont orthogonales.

Théorème

Soit $d_1, \ldots, d_k$ un ensemble de directions Q-conjuguées. Les vecteurs $d_1, \ldots, d_k$ sont linéairement indépendants.

Corollaire

Soit $Q \in \mathbb{R}^{n \times n}$, définie positive. Le nombre maximal de direction Q-conjuguées est n.

Idée de la méthode : optimiser successivement sur les n directions conjuguées.

$$x_{k+1} = x_k + \alpha_k d_k \qquad k = 1, \ldots, n$$

avec

$$\alpha_k = arg \min_{\alpha} f(x_k + \alpha d_k) \tag{2}$$

1/ Comment calculer le pas ?

D'après (2), on peut dériver en fonction de α :

$$\begin{array}{rcll} \frac{\partial f}{\partial \alpha} & = & d_k^t \nabla f(x_k + \alpha_k d_k) & = & 0 \\ & = & d_k^t (Q(x_k + \alpha_k d_k) + b) & = & 0 \\ & = & d_k^t Q x_k + \alpha_k d_k^t Q d_k + d_k^t b & = & 0 \end{array}$$

$$\Rightarrow \alpha_k = -\frac{d_k^t (Q x_k + b)}{d_k^t Q d_k}$$

2 / Comment calculer la direction ?

sans démonstration :

$$\beta_{k+1} = \frac{(Qx_{k+1} + b)^t (Qx_{k+1} + b)}{(Qx_k + b)^t (Qx_k + b)}$$

$$d_{k+1} = -Qx_{k+1} - b + \beta_{k+1} d_k$$

- méthode très performante : convergence en n itérations (au plus n directions).
- à l'instant k, le gradient est orthogonal à toutes les directions de descente précédente.

Méthode de Newton

- Utilisée pour trouver les zéros d'une fonction.
- Basée sur l'approximation linéaire.

Modèle linéaire mono-dimensionnel

On construit un modèle linéaire $m_{\hat{x}}(x) = f(\hat{x}) + (x - \hat{x})f^{'}(\hat{x})$ d'une fonction non-linéaire $f(x)$.
Approximation de $f(x + \triangle x) = 0$, par $f(x) + f^{'}(x).\triangle x = 0$.
Si $\triangle x = x^* - x \Rightarrow x^* = x - \frac{f(x)}{f^{'}(x)}$

Modèle linéaire en multi-dimensionnel

On construit un modèle linéaire $m_{\hat{x}}(x) = f(\hat{x}) + \nabla f(\hat{x})^t(x - \hat{x})..$
Approximation de $f(x + \triangle x) = 0$ par $f(x) + \nabla f(\hat{x})^t.\triangle x = 0$.
Si $\triangle x = x_{k+1} - x_k \Rightarrow x_{k+1} = x_k - \nabla f(x_k)^{-t}.f(x_k)$

- Méthode très performante (convergence quadratique)
- Mais peut diverger si :
  - point de départ trop éloigné de la racine
  - fonction trop non-linéaire
  - la dérivée de f à la solution est proche de zéro

Méthode de Newton pour l'optimisation

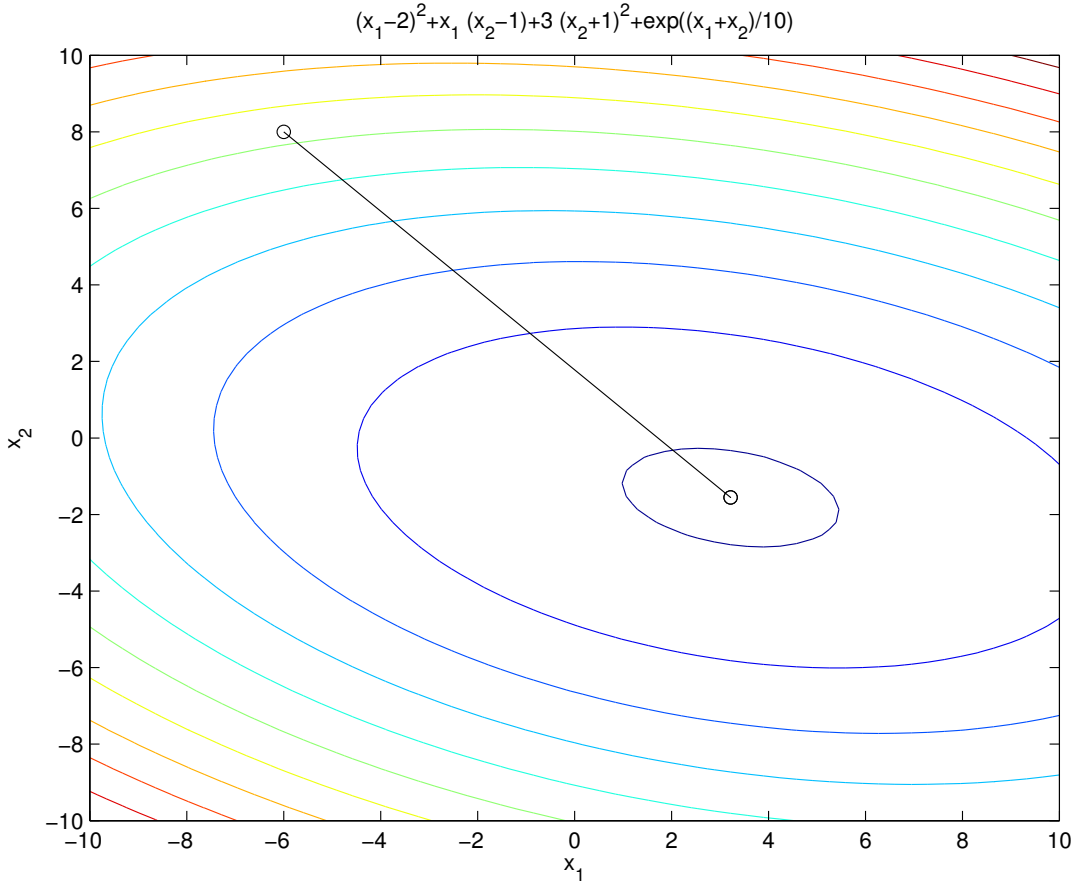
Dans le cas de l'optimisation locale d'une fonction, on s'intéresse à trouver un point x tel que $\nabla f(x) = 0$.
L'approximation devient :

$$\nabla f(x) + \nabla^2 f(x) \triangle x = 0 \Rightarrow x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

$d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$ est appelée direction de Newton.

- d_k est solution de l'approximation au second ordre de f au voisinage de x_k : $\arg\min_{d \in \mathcal{R}^n} f(x_k) + \nabla f(x_k)^t.d + \frac{1}{2} d^t.\nabla^2 f(x_k).d$
- si $\nabla^2 f(x_k)$ est déf. positive, c'est bien une méthode de descente à pas fixe.

Même exemple que pour le gradient.
2 itérations, $\|f(x_{k+1}) - f(x_k))\| \leq 10^{-8}$.
Beaucoup plus performant que les méthodes de gradient !!!



- méthode très performante (convergence quadratique)
- peut diverger si point de départ trop éloigné
- difficulté et coût de calcul de $\nabla^2 f(x_k)$.
- ne fonctionne pas si $\nabla^2 f(x_k)$ pas inversible.
- résout uniquement $\nabla f(x) = 0$, pas de garantie d'optimalité
- A la base de nombreuses autres méthodes.
  $\rightarrow$ On peut construire $\nabla^2 f(x)$ par un processus itératif. C'est la méthode de quasi-Newton.

L'idée est de calculer remplacer l'équation itérative par :
$x_{k+1} = x_k - \alpha_k . H_k^{-1} . \nabla f(x_k)$
avec

- la matrice H_k une bonne approximation de la hessienne $\nabla^2 f(x)$
- $\alpha > 0$ un pas calculé par une recherche linéaire

Méthode de quasi-Newton (zéro d'une fonction)

Dans certains cas le calcul des dérivées peut être impossible (forme analytique indisponible) ou bien trop coûteux en temps de calcul.
On utilise alors une approximation des dérivées : $a_k = \frac{f(x_k+s)-f(x_k)}{s}$
Si l'on cherche a annuler une fonction $f(x)$: $x_{k+1} = x_k - \frac{f(x_k)}{a_k}$

Modèle linéaire sécant d'une fonction

Soient f une fonction continue de n variables ($f : \mathbb{R}^n \to \mathbb{R}^m$) et $A \in \mathbb{R}^{m\times n}$.
Le modèle linéaire sécant de f en $\hat{x}$ est une fonction
$m_{\hat{x},A}(x) = f(\hat{x}) + A(x - \hat{x})$
A : approximation de la Jacobienne en $\hat{x}$.

Equation sécante

Soit : $d_{k-1} = x_k - x_{k-1}$ et $y_{k-1} = f(x_k) - f(x_{k-1})$
Un modèle linéaire vérifie l'équation sécante en x_k et x_{k-1} si
$A.d_{k-1} = y_{k-1}$.

Méthode de quasi-Newton (zéro d'une fonction)

Etant donné x_k, x_{k-1}, $f(x_k)$ et $f(x_{k-1})$ comment calculer la matrice A qui vérifie l'équation de la sécante ?
n équations pour n^2 inconnues : infinité de solutions.
Idée de Broyden : choisir le modèle linéaire le plus proche du modèle établi lors de l'itération précédente afin de conserver ce qui a déjà été calculé.

Résultat de Broyden

Soit $m_{\hat{x},A}(x)$ le modèle linéaire sécant de f en x_{k-1}. Soit $x_k \neq x_{k-1}$.
Alors le modèle linéaire sécant de f en x_k peut s'écrire :
$m_{x_k,A_k}(x) = f(x_k) + A_k(x - x_k)$
avec : $A_k = A_{k-1} + \frac{(y_{k-1} - A_{k-1}.d_{k-1}).d_{k-1}^t}{d_{k-1}^t.d_{k-1}}$

Méthode de Quasi-Newton en optimisation (zéro d'un gradient)

Comment calculer une bonne approximation de $\nabla^2 f(x_{k+1})$, H_{k+1} (ou de de $\nabla^2 f(x_{k+1})^{-1}$, B_{k+1}) connaissant x_k, x_{k+1}, $\nabla f(x_k)$ et $\nabla f(x_{k+1})$?

Lorsqu'on cherche a annuler le gradient on a :

$$\nabla f(x) + \nabla^2 f(x) \triangle x = 0 \Rightarrow x_{k+1} = x_k - H_k^{-1} \nabla f(x_k)$$

Plutôt que de calculer à chaque itération l'approximation de la Hessienne $H(x_k)$, on peut utiliser : Broyden-Fletcher-Goldfarb-Shanno (BFGS) :

$$H(x_{k+1}) = H(x_k) + \frac{y_k . y_k^t}{y_k^t . d_k} - \frac{H_k . d_k . d_k^t . H_k}{d_k^t . H_k . d_k}$$

$$B(x_{k+1}) = (I - \frac{d_k . y_k^t}{y_k^t . d_k})^t B(x_k)(I - \frac{d_k . y_k^t}{y_k^t . d_k}) + \frac{d_k . d_k^t}{y_k^t . d_k}$$

avec

$d_k = x_{k+1} - x_k$ et $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$

H_0 symétrique définie positive

3 Moindres carrés

Soit (P_{mc}) le problème d'optimisation suivant

$$min\ f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x) = \frac{1}{2}\|r(x)\|_2^2$$

avec $x \in \Re^n$, $r : \mathbb{R}^n \to \mathbb{R}^m$ et les r_j sont des fonctions différentiables. On suppose $m \geq n$.

Ce type de problème se rencontre fréquemment en identification de paramètres, en fouille de données, en problème inverse,....
En général les variables d'optimisation du vecteur x sont les paramètres du modèle proposé. On effectue m mesures et on cherche les x_i qui permettent d'ajuster au mieux le modèle aux mesures.
Remarquons que si $r(x) = 0$ a des solutions alors ce sont aussi des solutions de (P_{mc})

Résidus

Soit le vecteur résidus $r(x) = (r_1(x), r_2(x), ..., r_m(x))^t$, fonction de $\mathcal{R}^n \rightarrow \mathcal{R}^m$

Alors la matrice jacobienne $J(x)$ de dimension mxn correspond aux $[\frac{\partial r_j}{\partial x_i}]_{i,j}$

$$\nabla f(x) = \sum_{j=1}^{m} r_j(x).\nabla r_j(x) = J(x)^t.r(x)$$

$$\nabla^2 f(x) = \sum_{j=1}^{m} \nabla r_j(x).\nabla r_j(x)^t + \sum_{j=1}^{m} r_j(x).\nabla^2 r_j(x)$$

$$\nabla^2 f(x) = J(x)^t.J(x) + \sum_{j=1}^{m} r_j(x).\nabla^2 r_j(x)$$

3. 1 Moindres carrées linéaires

On se place dans la cas ou $r(x)$ est linéaire : $r(x) = A.x - b$ avec $A \in \mathcal{R}^{mxn}$.
On cherche à résoudre le problème (P_{mcl}) : $min\ f(x) = \frac{1}{2}\|A.x - b\|_2^2$

La Jacobienne en tout x est $J(x) = A$
$\nabla f(x) = A^t.(A.x - b)$ et $\nabla^2 f(x) = H_f(x) = J(x)^t.J(x) = A^t.A$
La matrice $H_f(x)$ est positive en tout x donc le problème des moindres carrées linéaire est convexe.

Calcul des points critiques

On cherche à vérifier les CN1, $\nabla f(x) = 0$

Equations normales

$A^t.A.x = A^t.b$

$x^\star$ est solution du problème $P_{mcl} \iff A^t.A.x^\star = A^t.b$
$x^* = (A^t.A)^{-1}.A^t.b$

Si A est de rang plein alors la fonction f est strictement convexe et x^* est l'unique minimum.

Résolution par la méthode de Newton

Appliquons la méthode de Newton avec $d_k = x_{k+1} - x_k$:
$A^t.A.d_k = -A^t.(A.x_k - b)$

$A^t.A.x_{k+1} = A^t.b$ pour tout x_k (on retrouve les équations normales du problème).
La méthode de Newton identifie la solution en une itération lorsque la fonction r est linéaire.

Exercice

Pouvez-vous calculer les paramètres d'un cercle (a, b, R) connaissant un ensemble de mesures (x_i, y_i) de points du cercle dans le plan ?

Moindres carrées linéaires récursif

Algorithme itératif qui utilise le calcul de x_k pour calculer la nouvelle estimé du vecteur x_{k+1} sans inversion de matrice (complexité en n^2 au lieu de n^3).

Moindres carrées linéaires pondérés

Il est parfois intéressant de remplacer la fonction à minimiser $f(x) = \frac{1}{2}\|A.x - b\|_2^2 = \frac{1}{2} r(x)^t.r(x)$ par $f(x) = \frac{1}{2} r(x)^t.Q.r(x)$ avec Q matrice symétrique définie positive (toutes les val. propres > 0).

Par exemple $Q = \begin{pmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & 0 & \lambda_n \end{pmatrix}$

avec $\lambda_1 > 0$, $\lambda_2 > 0$, ..., $\lambda_n > 0$ (ici on a fait n mesures)

On a maintenant :

$x^* = (A^t.Q.A)^{-1}.A^t.Q.b$

3. 2 Moindres carrées non-linéaires

On cherche à minimiser min $f(x) = \frac{1}{2}\|r(x)\|_2^2$ dans le cas ou $r(x)$ n'est plus linéaire par rapport à x.
On va exploiter les propriétés de la structure de $\nabla f(x)$ et $\nabla^2 f(x)$.

Méthode de Gauss-Newton

Idée : remplacer le problème des MCNL par un problème approché des MCL.
Soit le point x_k à l'itération k. Au voisinage de x_k on approxime notre problème par :

$$\min \tilde{f}(y) = \frac{1}{2}\|r(x_k) + J(x_k)(y - x_k)\|_2^2$$

On approxime $\nabla f(x)^2 = J(x)^t.J(x)$ (généralement prépondérant)

On a un problème des MCL qui vérifie les équations normales :
$J_k^t.J_k.(x_{k+1} - x_k) = -J_k^t.r_k$
La direction $d_k = x_{k+1} - x_k$ est appelée direction de Gauss-Newton.

Algorithm 1 : Algorithme de Gauss-Newton

$k = 0$
while *Critère d'arrêt* **do**
    calculer d_k solution de $J_k^t.J_k.d_k = -J_k^t.r_k$
    $x_{k+1} = d_k + x_k$
    $k = k + 1$
end
output : x_k

Avantages/Inconvénient

- Pas de calcul des dérivées secondes avec cette approximation (pour beaucoup de problèmes cette approximation est correcte)
- Si $J(x_k)$ est de rang plein et si $\nabla f(x_k) \neq 0$ alors la direction de Gauss-Newton est une direction de descente.
- Si $\nabla^2 r(x)$ est nulle à la solution, la convergence est quadratique.
- Sous certaines conditions on peut avoir une convergence globale vers un point critique de f.
- Si $J(x_k)$ n'est pas de rang plein on n'a pas de résultat de convergence.

Complément : Régularisation

Problème bien posé :

- la solution existe
- la solution est unique
- la solution dépend continûment des données.

Régulariser un problème mal posé, c'est le remplacer par un autre, bien posé, de sorte que l' "erreur commise" soit compensée par un gain de stabilité.

Méthode de Tikhonov

Si le problème $A.X = y$ est mal posé on le régularise par une (des) contrainte(s) $F(x) = 0$:

$$\min \frac{1}{2}\|A.x - b\|^2 + \frac{\epsilon^2}{2}\|F(x)\|^2 \qquad \min \frac{1}{2}\|A.x - b\|^2 + \frac{\epsilon^2}{2}\|x - x_0\|^2$$

Le choix "optimal" du paramètre de régularisation est "délicat".

Méthode de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt peut être vu comme une régularisation de l'algorithme de Gauss-Newton, en particulier lorsque la jacobienne de F n'est pas de rang plein. Mais il existe un lien étroit avec les méthodes dites de région de confiance.

Idée : comme dans la méthode de Gauss-Newton, on remplace le problème initial au voisinage de x_k par :

$$\min_{y \in \mathbb{R}^n} \frac{1}{2} \|r(x_k) + J(x_k)(y - x_k)\|_2^2$$

sous la contrainte $\|y - x_k\|_2^2 \leq \Delta_k$ (Δ_k rayon de la région de confiance).

On cherche à minimiser :

$$\min_{y \in \mathbb{R}^n} \frac{1}{2} \|r(x_k) + J(x_k)(y - x_k)\|_2^2 + \frac{1}{2} \lambda (\|y - x_k\|_2^2 - \Delta_k^2)$$

(Lagrangien - section **??**)

Les CN1 d'optimalité cette nouvelle fonction sont :

- $(J(x_k)^t.J(x_k) + \lambda I).d_k = -J(x_k)^t.r(x_k)$
- $\lambda = 0$ ou $\|y - x_k\| = \Delta_k$
- $\lambda \geq 0$

- Si $\|y - x_k\| < \Delta_k$ alors $\lambda = 0$, la contrainte est inactive, on n'atteint pas le bord de Δ_k alors on a une itération de Gauss-Newton.
- Sinon la solution est au bord de la région de confiance, $\|y - x_k\| = \Delta_k$. La matrice $(J(x - k)^t.J(x_k) + \lambda I)$ est maintenant définie positive et le pas effectué est bien un pas de descente de la fonction $f(x_k)$.
- $\lambda \geq 0$ peut être choisi fixe variable suivant qualité du pas.

Algorithm 2 : Algorithme de Levenberg-Marquardt

1 $k = 0$
2 **while** *Critère d'arrêt* **do**
3 | calculer d_k solution de $(J_k(x_k)^t.J_k(x_k) + \lambda.I).d_k = -J_k^t.r_k$
4 | $x_{k+1} = d_k + x_k$
5 | Mise à jour de λ
6 | $k = k + 1$
7 **end**
8 output : x_k

Part II - Optimisation non linéaire avec contraintes
A/ Conditions théoriques

Soit le problème d'optimisation $(P1)$ suivant

$$\min_{x \in \mathbb{X}} \; f(x)$$

$$h_i(x) = 0 \qquad i = 1, \ldots, m$$

$$g_j(x) \leq 0 \qquad j = 1, \ldots, p$$

où f, g et h sont différentiables sur $\mathcal{X}$.

Exemple : $\min_{x \in \mathcal{R}^n} \; f(x) = x^2$ sous $x \geq 1$
Condition sur le gradient ?
Directions de descente => directions admissibles

Résultats théoriques

Un point $x \in \mathbb{R}^n$ est admissible s'il vérifie toutes les contraintes.

Soit x un point admissible pour le problème général d'optimisation $(P1)$.
Une direction d est dite admissible en x s'il existe $\epsilon > 0$ tel que $x + \alpha d$ est admissible pour $0 \leq \alpha \leq \epsilon$

Conditions nécessaires d'optimalité pour des contraintes générales

Soit x^* un minimum local du problème général d'optimisation $(P1)$ alors
$\nabla f(x^*)^t.d \geq 0$
pour toute direction d admissible (à la limite) en x^*.

Contraintes actives

Pour $1 \leq j \leq p$, une contrainte $g_j(x) \leq 0$ est dite active en x^* si $g_j(x^*) = 0$ et inactive en x^* si $g_j(x^*) < 0$
Intérêt ?

Soit x^+ un point admissible pour le problème général d'optimisation $(P1)$.

Cône des directions

On appelle cône des directions en x^+, $CD(x^+)$, l'ensemble constitué des directions d (et de leurs multiples) telles que :
$d^t.\nabla g_i(x^+) \leq 0, \quad \forall i = 1, ..., p$ tel que $g_i(x^+) = 0$
et $d^t.\nabla h_i(x^+) = 0, \quad \forall i = 1, ..., m$.

Indépendance linéaire des contraintes

Les contraintes sont linéairement indépendantes en x^+ si les gradients des contraintes égalités, $\nabla h(x^+)$, et les gradients des contraintes d'inégalités actives en x^+, $\nabla g_i(x^+)$, sont linéairement indépendants.

Qualification des contraintes

La qualification des contraintes est vérifiée si $\forall d \in CD(x^+)$ d est une direction admissible à la limite en x^+.
Les directions admissibles à la limite sont une extension de la notion de direction admissible difficile à calculer. Nous supposerons que les contraintes sont qualifiées si elles sont linéairement indépendantes.

Optimisation avec contrainte égalité

Soit $(P2)$ le problème d'optimisation suivant

$$\min_{x \in \mathcal{X}} \; f(x)$$

$$h(x) = 0$$

où f et h sont différentiables sur $\mathcal{X}$.

Fonction lagrangienne L

La fonction L définie par

$$L(x, \lambda) = f(x) + \lambda^t h(x)$$

est appelée lagrangien ou fonction lagrangienne du problème $(P2)$.
avec λ multiplicateurs de Lagrange.

Théorème (condition nécessaire de Karush-Kuhn-Tucker)

Soit x^* un minimum local du problème $(P2)$ alors

$$\begin{cases} \nabla_x L(x^*, \lambda) &= 0 \\ \nabla_\lambda L(x^*, \lambda) &= 0 \end{cases}$$

On peut écrire ce système :

$$\nabla_{x,\lambda} L(x^*, \lambda) = 0$$

ou encore plus simplement

$$\nabla L(x^*, \lambda) = 0$$

Si le système comporte n inconnus et p contraintes, le théorème précédent nous fournit $n + p$ équations (et on a $n + p$ contraintes d'égalité).

Théorème

Soit x^* un minimum local du problème $(P2)$:
Si les contraintes sont linéairement indépendantes en x^* alors il existe un vecteur unique λ^* tel que $\nabla L(x^*, \lambda) = 0$.
Si f et h sont deux fois différentiables alors

$$y^t.\nabla^2_{xx} L(x^*, \lambda^*).y \geq 0, \forall y \in \mathcal{D}(x^*)$$

avec $\mathcal{D}$ cône des directions admissibles.
De plus on a :

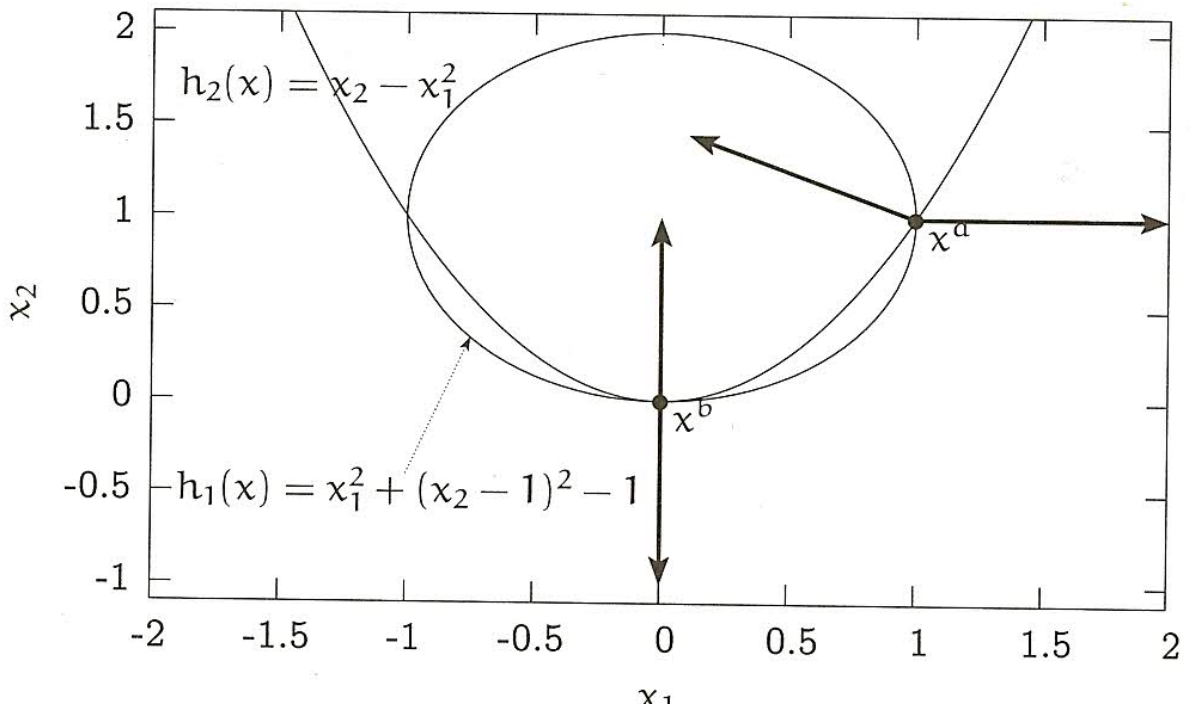
$$\lambda^* = -(\nabla h(x^*)^t.\nabla h(x^*))^{-1}.\nabla h(x^*)^t.\nabla f(x^*)$$

(tableau)

Exemple

$$f(x_1, x_2) = x_1 + x_2$$

$$\text{s.c. } x_1^2 + (x_2 - 1)^2 = 1 \text{ et } -x_1^2 + x_2 = 0$$



Exemple 4

$$f(x) = -x_1x_2 - x_1x_3 - x_2x_3$$

$$\text{s.c. } x_1 + x_2 + x_3 = 3.$$

Les conditions de KKT, se prêtent très bien aux fonctions objectifs quadratiques et aux contraintes linéaires

$$\min f(x) = \frac{1}{2} x^t Q x + g^t x$$

$$\text{s.c. } Ax = b.$$

$$\begin{aligned} \nabla_x f(x) &= Qx + g \\ \nabla_x h(x) &= -A^t \end{aligned}$$

$$\begin{aligned} \nabla_x L(x,\lambda) &= Qx + g - A^t\lambda &= 0 \\ \nabla_\lambda L(x,\lambda) &= b - A.x &= 0. \end{aligned}$$

On résout un système linéaire.
Donner les valeurs de x^* et λ^*.

Optimisation avec contrainte égalité et inégalité

Soit $(P3)$ le problème d'optimisation suivant

$$\min_{x \in \mathcal{X}} \; f(x)$$

$$h_i(x) = 0 \qquad i = 1, \ldots, m$$

$$g_j(x) \leq 0 \qquad j = 1, \ldots, p$$

où f, g et h sont différentiables sur $\mathcal{X}$.

Le lagrangien s'écrit désormais

$$L(x, \lambda, \mu) = f(x) + \lambda^t h(x) + \mu^t g(x)$$

Théorème(conditions nécessaires de Karush-Kuhn-Tucker)

Soit x^* un minimum local du problème $(P3)$:
Si les contraintes sont linéairement indépendantes en x^* alors il existe un vecteur unique λ^* et un vecteur unique μ^* tels que :

$$\nabla_x L(x^*, \lambda^*, \mu^*) = 0,$$
$$\begin{array}{ll} \mu_j \geq 0 & j = 1, \ldots, p \\ \mu_j g_j(x^*) = 0 & j = 1, \ldots, p \end{array}.$$

Si f et h sont deux fois différentiables alors
$y^t.\nabla^2_{xx} L(x^*, \lambda^*, \mu^*).y \geq 0, \quad \forall y \neq 0$ tel que

$$\begin{array}{ll} y^t.\nabla h_i(x^*) = 0 & i = 1, \ldots, m \\ y^t.\nabla g_i(x^*) = 0 & i = 1, \ldots, p \mid g_i(x^*) = 0 \ (\textit{contraintes actives}) \end{array}.$$

Théorème(conditions suffisantes)

Soient f et g des fonctions deux fois différentiables.

Soient x^*, λ^* et μ^* tels que :

$$\nabla_x L(x^*, \lambda^*, \mu^*) = 0$$
$$h(x^*) = 0$$
$$g(x^*) \leq 0$$
$$\mu^* \geq 0 \qquad j = 1, \ldots, p$$
$$\mu_j^* g_j(x^*) = 0 \qquad j = 1, \ldots, p$$
$$\mu_j^* > 0 \qquad \forall \textit{ contrainte } j \textit{ active}$$

$y^t . \nabla^2_{xx} L(x^*, \lambda^*, \mu^*) . y > 0, \quad \forall y \neq 0$ tel que

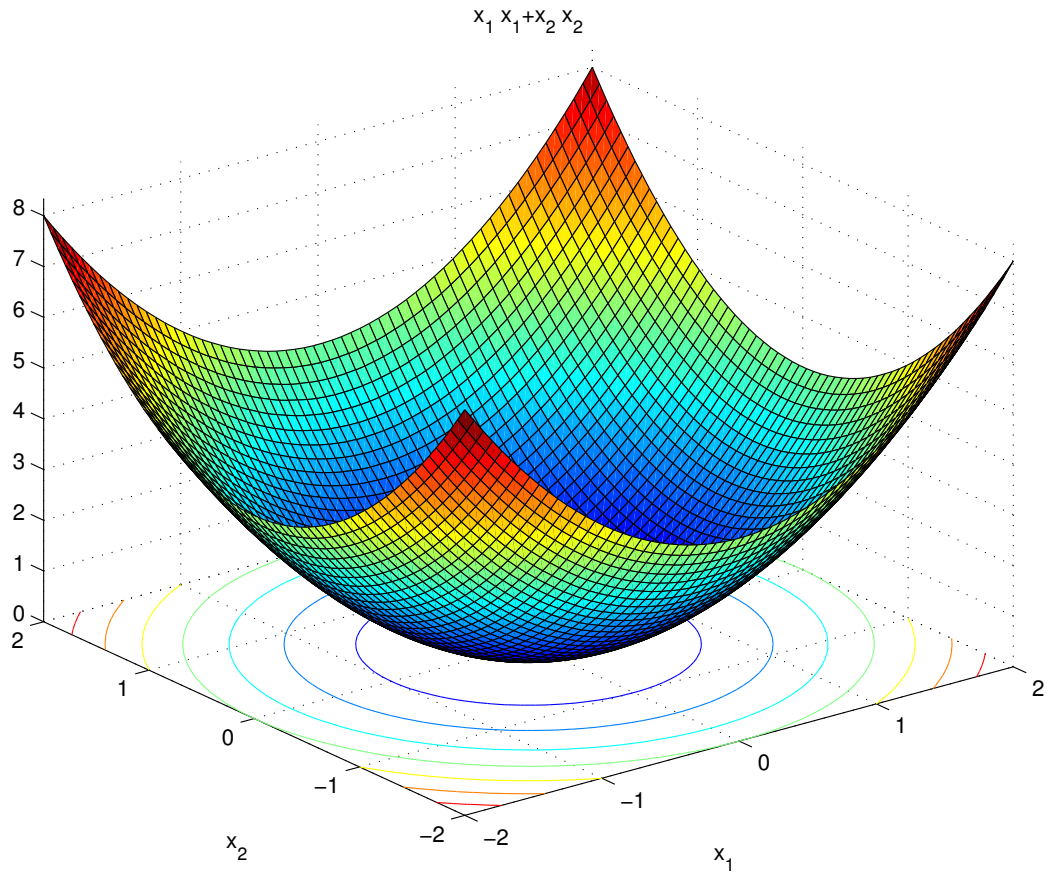
$$y^t . \nabla h_i(x^*) = 0 \quad i = 1, \ldots, m$$
$$y^t . \nabla g_i(x^*) = 0 \quad i = 1, \ldots, p \qquad \textit{tel que } \; g_i(x^*) = 0$$

Alors x^* est un minimum local.

Exemple 5

$$f(x_1, x_2) = x_1^2 + x_2^2$$

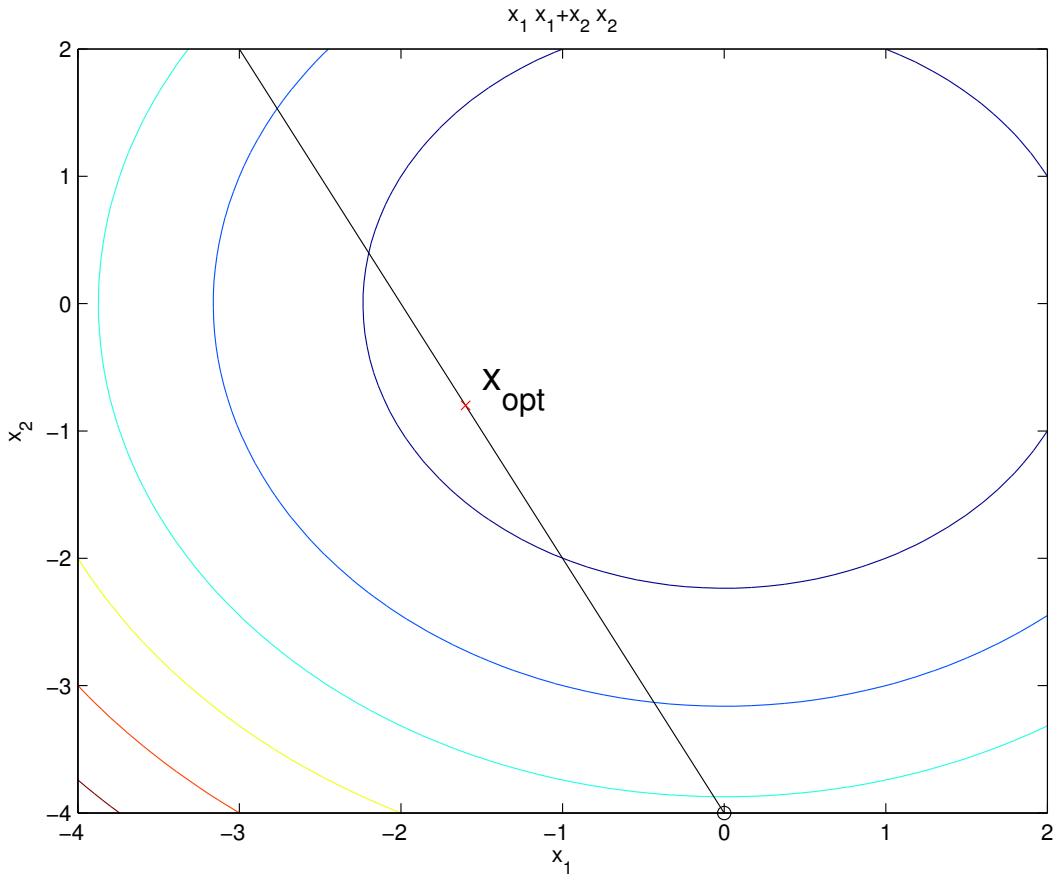
$$\text{s.c. } 2x_1 + x_2 \leq -4.$$



Exemple 5

$$f(x_1, x_2) = x_1^2 + x_2^2$$

$$\text{s.c. } 2x_1 + x_2 \leq -4.$$



Part IV - Optimisation avec contraintes
B/ Méthodes et algorithmes

Nous allons maintenant étudier les méthodes de recherche pour les problèmes contraints.

Problème :

$$\min_{x \in X \subseteq \mathbb{R}^n} f(x)$$

Nous allons réutiliser les principes des méthodes de la partie I ainsi que les propriétés théoriques de la partie II.

L'optimisation sous contraintes est généralement un problème difficile. Cependant, les conditions KKT ont une importance fondamentale pour l'analyse du problème et pour trouver des d'éventuels candidats à l'optimum.

Méthode du gradient projeté

Idée : suivre la direction de plus forte pente.

$$x_{k+1} = x_k + \alpha_k . d_k$$

$=>$ aucune garantie que le point est admissible

Quand le point est non admissible, faire une opération de projection, $P_k[.]$, sur $X \subseteq \mathbb{R}^n$.
$x_{k+1} = P_k[x_k + \alpha_k . d_k]$
Le critère d'optimalité pour un problème avec contraintes n'implique pas que le gradient est nul.
On ne peut pas utiliser la norme du gradient comme critère d'arrêt.
$\|x_{k+1} - x_k\| < \epsilon$

Algorithme du gradient projeté

Direction choisie : $\nabla f(x)$

Initiations : $k = 0$, x_0, α_0

Itération

1. Calculer $\nabla f(x_k)$
2. $x_{k+1} = P_k[x_k - \alpha_k . \nabla f(x_k)]$
3. $k = k + 1$

Critère d'arrêt : $\|x_{k+1} - x_k\| < \epsilon$

- A chaque itération il faut calculer l'opérateur de projection sur l'ensemble de contraintes K .
- Le calcul d'une projection est lui même un problème d'optimisation avec contrainte.
- Cette approche est donc intéressante seulement si le calcul de la projection peut être réalisé explicitement ou par un algorithme très rapide.

Lagrangien augmenté

Soit $(P2)$ le problème d'optimisation suivant

$$\min_{x \in \mathcal{X}} \; f(x)$$

$$h(x) = 0$$

où f et h sont différentiables sur $\mathcal{X}$.

Idée : transformer le problème avec contraintes en une suite de problèmes sans contraintes, en pénalisant de plus en plus la violation (éventuelle) des contraintes. On s'autorise à ne pas respecter les contraintes et à restaurer l'admissibilité en itérant le processus.

Lagrangien augmenté

On appelle lagrangien augmenté la fonction L_c définie par

$$L_c(x, \lambda) = f(x) + \lambda^t h(x) + \frac{c}{2}\|h(x)\|^2$$

Si c est grand et $h(x) \neq 0$ alors $\frac{c}{2}\|h(x)\|^2$ est grand $\Longrightarrow$ on est loin du minimum de $L_c(x, \lambda)$.

$$\nabla_x L_c(x, \lambda) = \nabla f(x) + \nabla h(x)\lambda + c\nabla h(x)h(x)$$

On cherche à résoudre une suite de problème d'optimisation sans contrainte pour chaque λ donné et c donné.
A chaque problème (à chaque itération), λ et c vont évoluer.

$$x_k = \arg\min_{x \in \mathbb{R}^n} \; L_{c_k}(x, \lambda_k)$$

Approximation des multiplicateurs de Lagrange

Soit la suite (c_k), $c_k \in \mathbb{R}$ et $0 < c_k < c_{k+1}$ avec $\lim_{k\to\infty} c_k = +\infty$,
soit la suite bornée (λ_k), $\lambda_k \in \mathbb{R}^m$,
soit la suite (ϵ_k), avec $0 < \epsilon_k$ et $\lim_{k\to\infty} \epsilon_k = 0$,
soit la suite (x_k) telle que $\|\nabla_x L_{c_k}(x_k, \lambda_k)\|^2 \leq \epsilon_k$
S'il existe une sous-suite $(x_k)_{k\in K}$ de (x_k) qui converge vers l'optimum x^* et si $\nabla h(x^*)$ est de rang plein alors :

$$\lim_{k\to\infty} \lambda_k + c_k.h(x_k) = \lambda^*$$

et x^* et λ^* vérifient les CN d'optimalité du premier ordre :
$\nabla f(x^*) + \nabla h(x^*)\lambda^* = 0$ et $h(x^*) = 0$

Chercher $x_k = arg\min_{x\in\mathbb{R}^n} L_{c_k}(x, \lambda_k)$ avec méthode Newton, ...
Tester $\|h(x)\|^2$ pour mettre à jour les λ_k ou les c_k :

- si $\|h(x)\|^2$ petite alors λ_k : $\lambda_{k+1} = \lambda_k + c_k.h(x_k)$.
- si $\|h(x)\|^2$ grand alors c_k : $c_{k+1} = \tau.c_k$.

jusqu'à avoir $\|h(x_k)\|^2 \leq \epsilon$ et $\|\nabla_x L_{c_k}(x_k, \lambda_k)\|^2 \leq \epsilon$.