M2
Optimisation
Concepts de base et méthodes

André Crosnier

6 octobre 2017

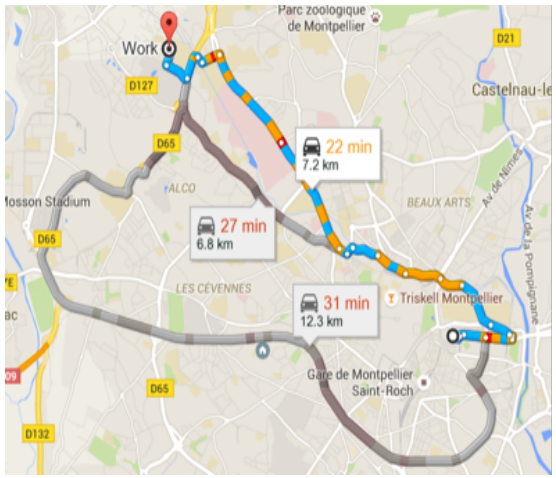
Objectifs et contenu

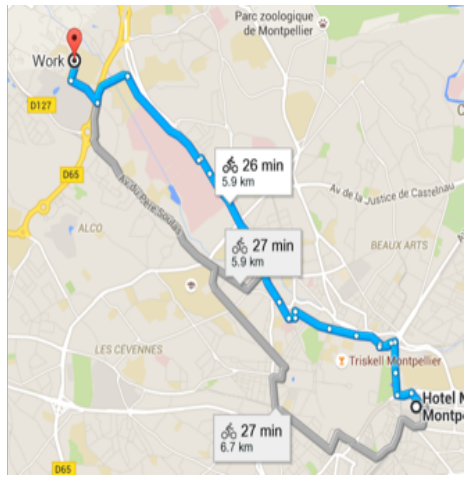
Objectifs du cours : étudier les concepts de base utilisés en **optimisation continue**, comprendre les méthodes, et résoudre quelques problèmes simples

- Partie 1 : Introduction
- Optimisation en dimension 1 (cf. annexes)
- Partie 2 : Optimisation non linéaire sans contraintes
- Partie 3 : Optimisation non linéaire avec contraintes

Quoi ?

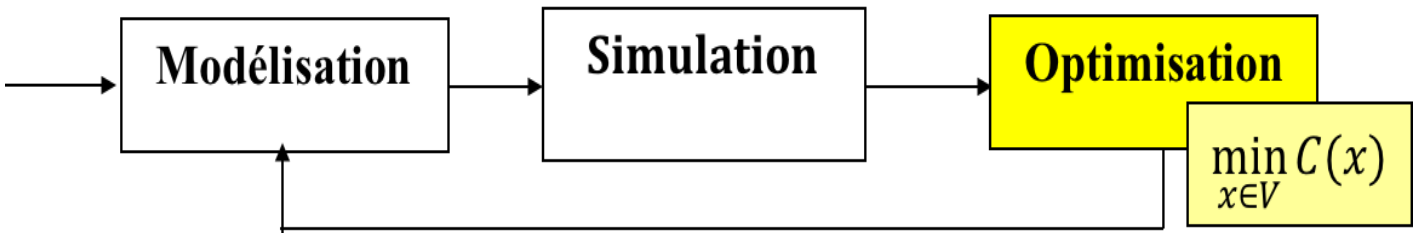
- Proposer des méthodes permettant de résoudre le problème général suivant : trouver une(des) solutions(s) la(les) meilleure(s) possible(s) afin d'optimiser un objectif (coût) donné.
- Sujet très ancien qui connaît un nouvel essor (très important) depuis l'apparition des ordinateurs, et qui ne cesse de prendre de l'importance en particulier avec le développement des systèmes embarqués





Quoi ?

- Les méthodes isues de l'optimisation s'appliquent dans de très nombreux secteur d'activités : économie, gestion, planification, logistique, électronique, automatique/robotique, mécanique, ...



- La grandeur x est un vecteur composé des variables que l'on cherche à optimiser pour la fonction de coût $C(x)$. x est choisi à l'intérieur d'un espace de recherche V.

Quoi ?

- Deux grandes classes de problèmes :
  1. Optimisation continue : le nombre de solutions potentielles dans V est infini ($V = \mathbb{R}^n$: variables continues)
  2. Optimisation combinatoire : il existe un nombre fini de solutions potentielles qu'il suffit d'énumérer afin de déterminer la solution optimale (ou réalisable).
- L'optimisation continue est souvent plus facile à résoudre, sauf lorsque les fonctions de coût sont fortement non linéaires ou discontinues.
- L'approche combinatoire peut conduire à des problèmes d'optimisation "difficiles" (NP-complets) : utilisation de métaheuristiques.

Exemple 1 : livraison d'une marchandise

On cherche à optimiser la livraison de marchandise. On dispose de M entrepôts disposant chacun d'un stock s_i. Il faut livrer N clients qui ont commandé chacun une quantité r_j. Le coût unitaire du transport entre l'entrepôt i et le client j est égal à c_{ij}.

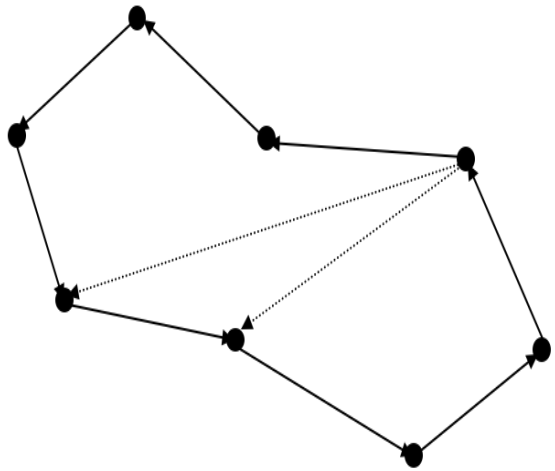
- Les variables de décision sont les quantités v_{ij} de marchandises partant de l'entrepôt i vers le client j.
- On veut minimiser le coût du transport tout en satisfaisant les clients :

$$C(v) = \min_{v_{ij}} \sum_{i=1}^{M} \sum_{j=1}^{N} c_{ij} v_{ij}$$

- Contraintes : $v_{ij} \geq 0$, $\sum_{j=1}^{N} v_{ij} \leq s_i$, $\sum_{i=1}^{M} v_{ij} = r_j$

Exemple 2 : voyageur de commerce

Un représentant de commerce doit visiter n villes et revenir à son point de départ en un temps minimum. Le temps pour rejoindre la ville i à la ville j est noté t_{ij}.



- On trace un graphe orienté des n villes reliées entre elles par des arcs pondérés par les temps t_{ij}
- Objectif : trouver un cycle dans le graphe qui passe une fois et une seule par toutes les villes et qui minimise le temps total.
- Problème d'optimisation combinatoire

Exemple 3 : commande optimale

On considère un système défini par la représentation d'état : $\dot{x} = Ax + Bu + b$. On cherche une loi de commande $u = -Kx$ qui minimise le critère de commande optimale suivant :

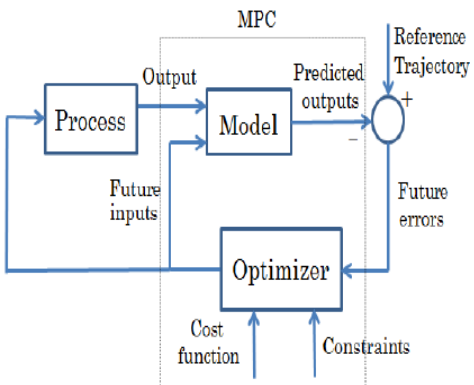
$$C(u) = \int_0^t (x^T Q x + u^T R u) dt$$

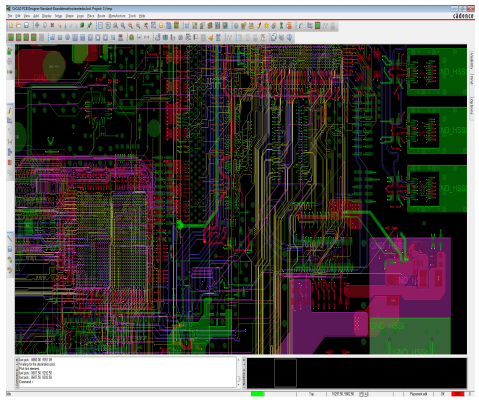
sous la contrainte de commande admissible $u(t) \in U \subset \mathbb{R}^M$

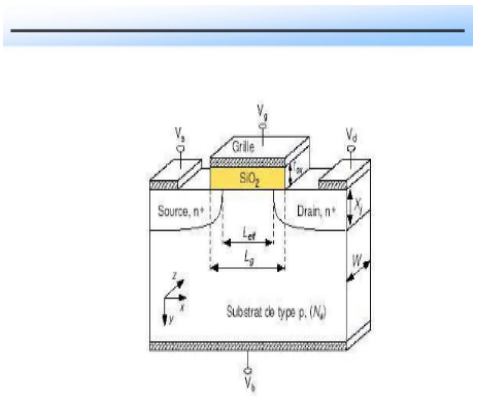
Pourquoi ?

Dans nos disciplines, l'optimisation joue un rôle important :

- Synthèse de lois de commande pour les systèmes : commande QP ou LP, commande LQR, commande prédictive (Model Predictive Control)
- Caractérisation et dimensionnement de composants en électronique
- Identification de systèmes (robot, composants)
- Placement-routage de circuits
- Optimisation de la consommation énergétique







Formalisation d'un problème d'optimisation

Un problème d'optimisation est défini par la donnée de trois éléments :

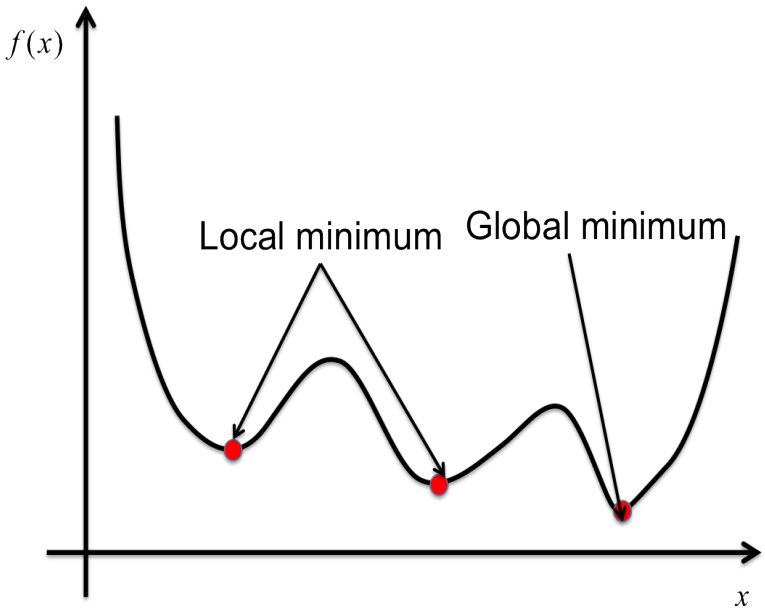
- **Une fonction de coût** (ou fonction objectif) f qui spécifie la grandeur que l'on veut optimiser : énergie, volume, surface, poids, ...
- **Un espace de recherche** V définissant un jeu de variables (variables de décision) $x = (x_1, \cdots, x_n)^T \in V$ que l'on cherche à déterminer afin d'optimiser la fonction de coût $f(x)$. x peut représenter des grandeurs de nature différentes (dimension, matériau, ...)
- **Un ensemble de contraintes** définies par des équations écrites sous forme d'inégalités et/ou d'égalités et que la solution x doit vérifier

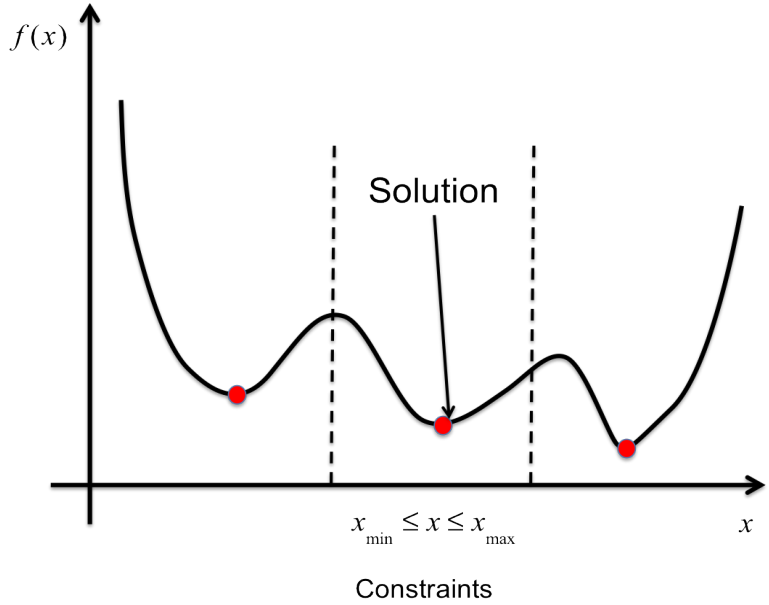
Le problème d'optimisation peut alors s'écrire sous la forme

$$\begin{aligned} &\min_x \quad f(x) \\ &\text{sous les contraintes} \\ &\qquad h_i(x) = 0 \quad i = 1, \cdots, m \\ &\qquad g_j(x) \leq 0, \quad j = 1, \cdots, k \end{aligned} \tag{1}$$

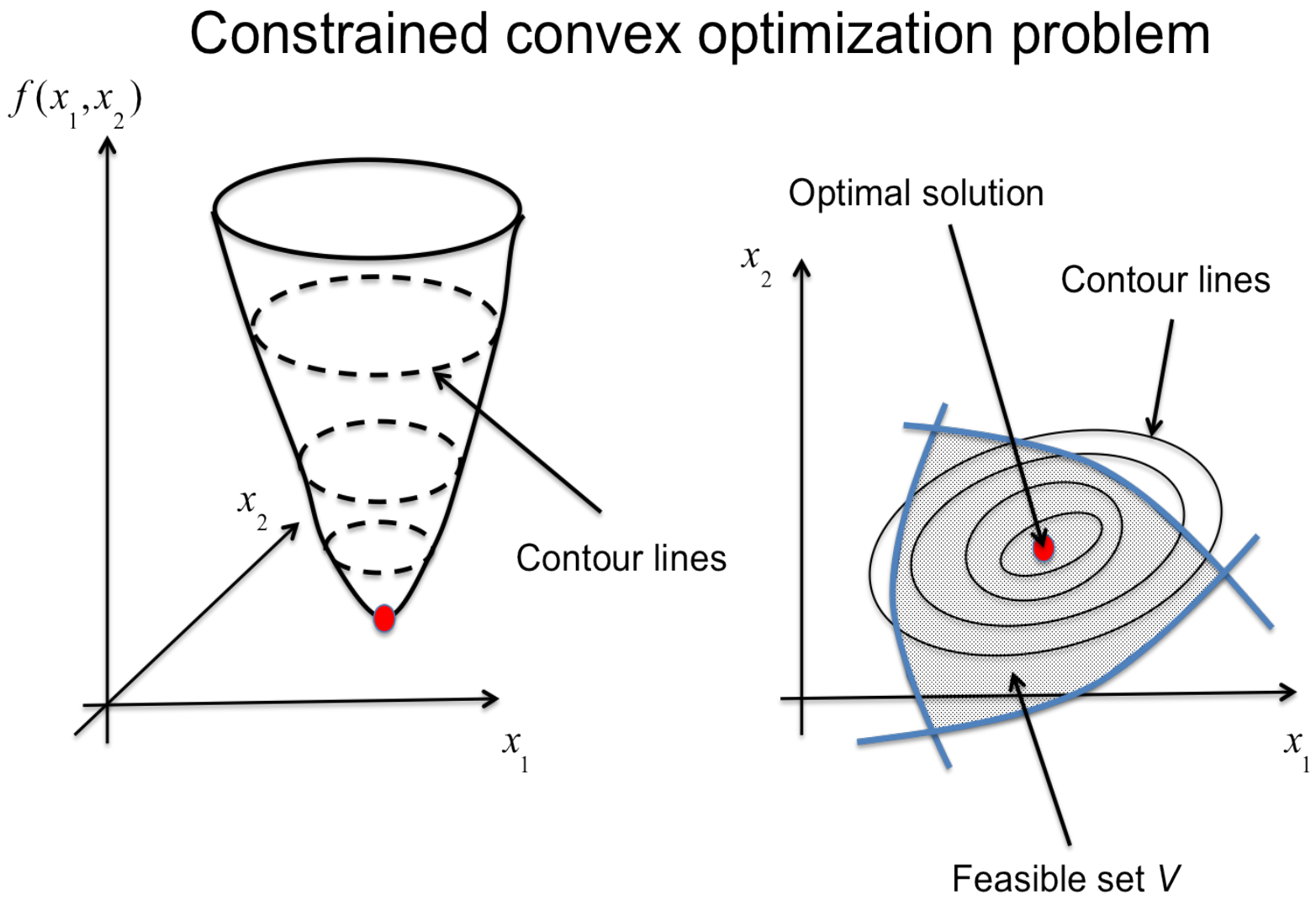
On cherche alors la solution optimale $x^* = \underset{x}{\text{argmin}} f(x)$.

Formalisation d'un problème d'optimisation





Formalisation d'un problème d'optimisation



Formalisation d'un problème d'optimisation

- Il existe de nombreuses techniques pour résoudre un problème d'optimisation. Ces techniques peuvent être classées en fonction de la nature de x, f et des contraintes. On considère la fonction de coût définie par :

$$\begin{array}{rccc} f: & V & \longrightarrow & \mathbb{R}^p \\ & x & \longrightarrow & f(x) \end{array} \tag{2}$$

- $V = \mathbb{R}^n$: **optimisation continue** ([1]) **sans contrainte**
- $V \subset \mathbb{R}^n$: **optimisation avec contraintes**. L'ensemble V est appelé ensemble des solutions admissibles et contient donc les solutions x qui satisfont les contraintes. Quand V est connu, l'optimisation se ramène à trouver la solution optimal x^* qui optimise la fonction $f(x)$
- $p > 1$: **optimisation multi-objectifs**. Un ensemble de fonctions de coût est alors défini. Par exemple, minimiser l'énergie consommée et la surface utilisée par un circuit

1. Cadre principal du cours

Formalisation d'un problème d'optimisation

- $f(x)$ est linéaire par rapport à x : **optimisation linéaire** (appelée aussi programmation linéaire). S'il existe des contraintes, elles doivent être aussi linéaires par rapport à x.
- $f(x)$ est non linéaire par rapport à x ou bien s'il existe au moins une contrainte non linéaire : **optimisation non linéaire**
- La fonction de coût $f_t(x(t))$ est fonction du temps : optimisation dynamique
- Si x est une variable aléatoire, optimisation stochastique

$\Rightarrow$ chaque cas fait appel à des techniques différentes. Il est donc nécessaire d'identifier la nature du problème d'optimisation à résoudre afin de choisir la bonne méthode de résolution.

Rappels

- Vecteur dans $\mathbb{R}^n$: $x = (x_1, \cdots, x_n)^T$
  - p-norme : $|| x ||_p = \left(\sum_{i=1}^{n} | x_i |^p\right)^{\frac{1}{p}}$
  - 2-norme : $|| x ||_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}}$ - distance Euclidienne
  - ∞-norme : $|| x ||_\infty = \max(x_1, \cdots, x_n)$
  - Produit scalaire : $< u, v >= u \bullet v = u^T v$

- Matrice dans $\mathbb{R}^{nxn}$: $A = [a_{ij}]_{i,j=1,\ldots,n}$
  - AA^T est une matrice symétrique
  - Si $\det A = 0$, A est singulière. Si $\det A \neq 0$, il existe une matrice inverse A^{-1} telle que $AA^{-1} = A^{-1}A = \mathbb{I}_n$. Si $\det A = 1$, A est une matrice orthonormée et $A^{-1} = A^T$
  - Une matrice symétrique A est définie positive (notée $A > 0$) si le produit scalaire $x^T Ax$ est positif pour tout vecteur x. Elle est définie semi-positive si $x^T Ax \geq 0$

Rappels

Soit f une fonction dérivable à valeur sur $\mathbb{R}$ telle que : $f : \mathbb{R}^n \longrightarrow \mathbb{R}$.

- Le gradient de f, noté ∇f est défini par : $\nabla f = \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n} \right)^T$.
  Le vecteur gradient en x est orthogonal à la ligne de contour passant par x.
- La matrice Hessienne de f est la matrice composée des dérivées secondes de f : $H = [\frac{\partial^2 f}{\partial x_i \partial x_j}]$. H est notée aussi $H = \nabla^2 f$
- Développement de Taylor au second ordre de f :

$$f(x+h) = f(x) + \nabla f(x)^T h + \frac{1}{2} h^T \nabla^2 f(x) h + \Theta(h) \qquad (3)$$

avec $\Theta(h) \to 0$ quand $|| h || \to 0$

Rappels

Exercise 1 : On considère la fonction de coût suivante :

$$f(x) = (x_1 - 1)^2 + 10(x_1^2 - x_2)^2 \tag{4}$$

- Etablir l'expression du gradient et de la matrice Hessienne de $f(x)$
- Trouver une solution x^* telle que $x^* = \underset{x}{\text{argmin}}\ f(x)$

Rappels

Convexité : c'est une propriété importante utilisée en optimisation.

- **Ensemble convexe :** Soit $S \subset \mathbb{R}^n$. L'ensemble S est convexe ssi

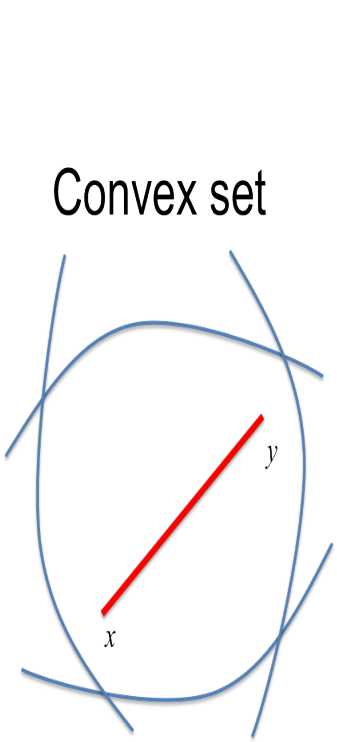
$$\forall x \in S, \forall y \in S, \forall \alpha \in [0,1], \quad \alpha x + (1-\alpha)y \in S \tag{5}$$

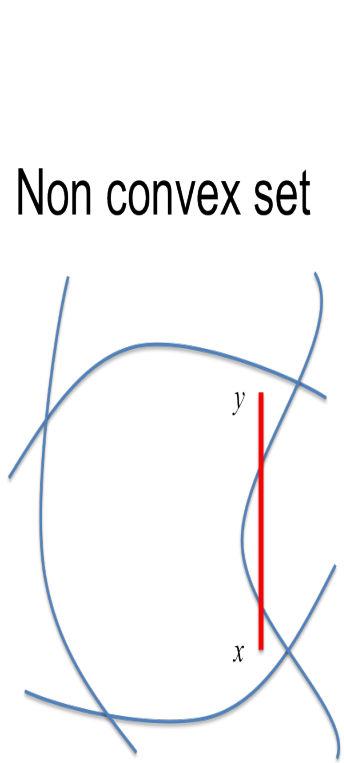
- **Fonction convexe :** on considère la fonction f telle que $f : \mathbb{R}^n \longrightarrow \mathbb{R}$. f est convexe ssi

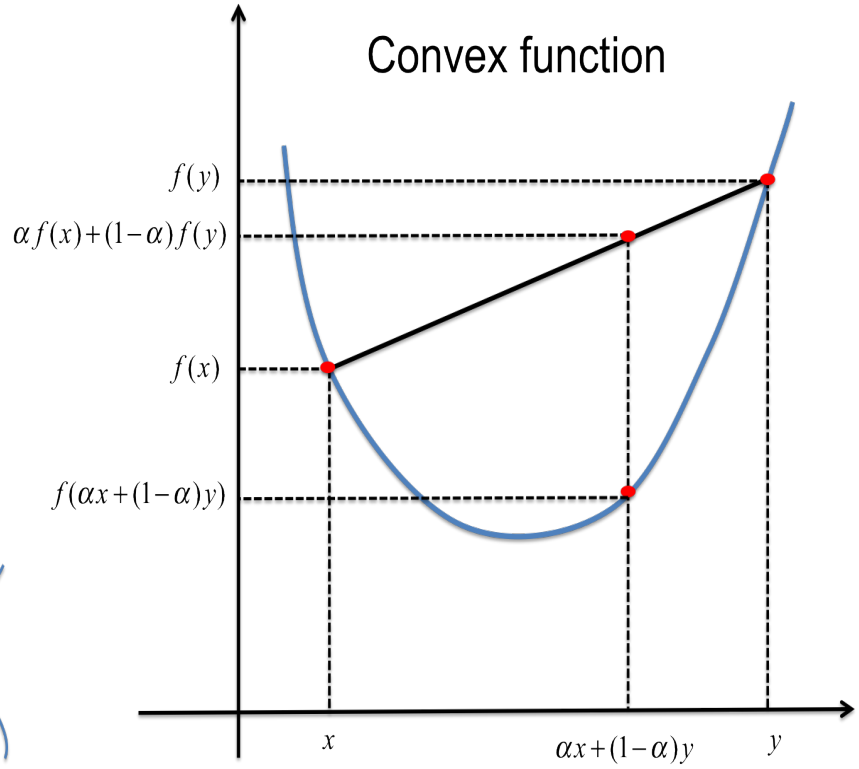
$$\begin{gathered} \forall x \in \mathbb{R}^n, \forall y \in \mathbb{R}^n, \forall \alpha \in [0,1] \\ f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) \end{gathered} \tag{6}$$

- **Propriétés des fonctions convexes :** les propriétés suivantes sont équivalentes :
  1. f est convexe
  2. $f(y) \geq f(x) + \nabla f(x)^T (y - x)$
  3. la matrice Hessienne de f est définie positive $\nabla^2 f > 0$. Les valeurs propres de $\nabla^2 f > 0$ sont toutes positives.
- **Théorème :** Si la fonction de coût f est convexe sur un ensemble convexe V, tout point de minimum local de f sur V est un minimum global. Lorsque f est strictement convexe alors il existe au plus un point de minimum.

Rappels


Convex set
y
x


Non convex set
y
x


Convex function
$f(y)$
$\alpha f(x)+(1-\alpha)f(y)$
$f(x)$
$f(\alpha x+(1-\alpha)y)$
x
$\alpha x+(1-\alpha)y$
y

Rappels

Exercise 2 : Démontrer que la fonction $f : \mathbb{R} \to \mathbb{R}$ définie par : $f(x) = x^2$ est strictement convexe.

$$\begin{gathered} \forall x \in \mathbb{R}, \forall y \in \mathbb{R}, x \neq y, \forall \alpha \in]0,1[\\ f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha) f(y) \end{gathered} \tag{7}$$

Rappels

Exercise 3 : On considère la fonction de coût suivante (forme quadratique) :

$$f(x) = \frac{1}{2} x^T A x - b^T x \tag{8}$$

avec A une matrice de dimension $n \times n$ et b un vecteur de dimension $n \times 1$.

- Etablir l'expression du gradient et de la matrice Hessienne de $f(x)$. Que devient le résultat quand A est symétrique ?
- Etablir l'expression analytique de la solution optimale x^* pour le problème : $\underset{x}{\text{optimize}}\ f(x)$.
- Soit f la fonction définie par : $f(x) = x_1^2 + 0.5x_2^2 - x_1 x_2 - x_1 - x_2$. Trouver A, b et x^*. Que représente la solution optimale (maximum ou minimum) ?

Rappels

Exercise 4 : On considère la fonction de coût suivante :

$$f(x) = || Ax - b ||_2^2 \tag{9}$$

avec A une matrice de dimension $n \times n$ et b un vecteur de dimension $n \times 1$.

Déterminer l'expression analytique de la solution optimale x^* pour le problème : $\underset{x}{\text{optimize}}\ f(x)$.

Rappels

Application : méthode des moindres carrés (linéaire)
On veut caractériser la droite $v = \alpha u + \beta$, où α et β sont deux inconnues, à partir d'un ensemble de mesures $(u_i, v_i), i = 1, \cdots, n$. Le problème peut être formalisé comme suit :

$$\min_{\alpha,\beta} \sum_{i=1}^{n} (v_i - \alpha u_i - \beta)^2 \tag{10}$$

Exercise 5 : Ecrire la fonction de coût sous la forme suivante : $f(x) = || Ax - b ||_2^2$. Donner les expressions de x, A et b

Optimization Toolbox dans Matlab : la fonction *lsqlin* résoud le problème

$$\min_x \frac{1}{2} || Cx - d ||_2^2 \tag{11}$$

avec les contraintes : $A\,x \leq b$, $A_{eq}\,x = b_{eq}$ et $lb \leq x \leq ub$

$x = lsqlin(C,d,A,b,Aeq,beq,lb,ub,x0)$

Section 2 : optimisation sans contraintes

Introduction

On considère la fonction de coût définie par :

$$\begin{array}{rccc} f: & \mathbb{R}^n & \longrightarrow & \mathbb{R} \\ & x & \longrightarrow & f(x) \end{array} \tag{12}$$

avec $n > 1$. On cherche une solution au problème : $\min_x f(x)$.

Exemple 1 : on considère la fonction : $f(x_1, x_2) = x_1^2 + x_2^2$. $f(x)$ est une forme quadratique définie par $f(x) = \frac{1}{2} x^T H x - b^T x$ avec $b = 0$ et

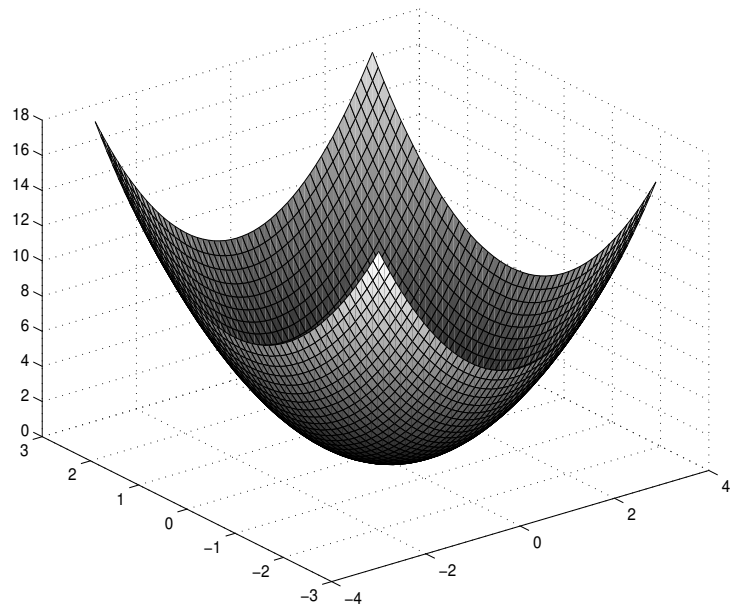
$$\nabla f = Hx - b = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} \tag{13}$$

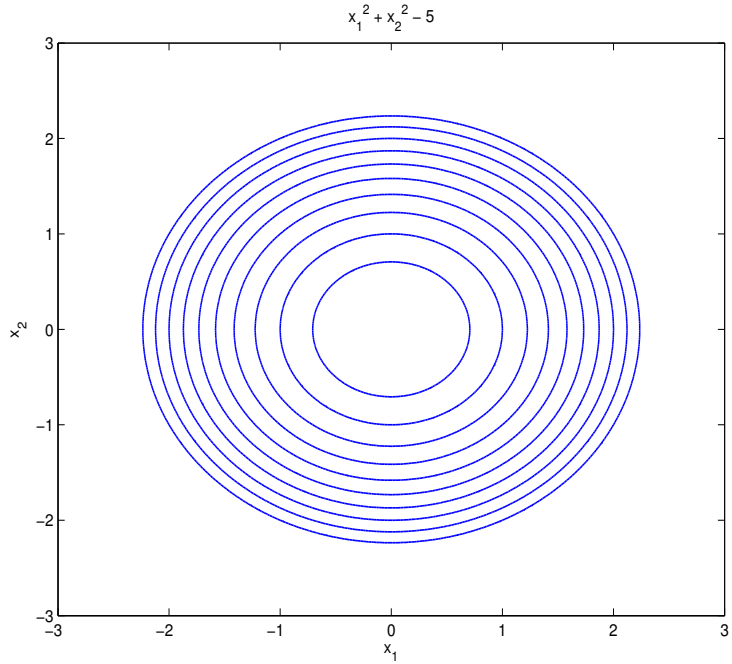
$$\nabla^2 f = H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \tag{14}$$

Introduction

H est une matrice définie positive ($x^T Hx > 0$) et donc $f(x)$ est convexe. La solution unique au problème d'optimisation est donnée par :

$$\nabla f(x) = 0 \Longleftrightarrow x^* = H^{-1}b = 0 \tag{15}$$





Introduction

Exemple 2 : moindres carrés non linéaires
En traitement d'image (ou en caractérisation), on cherche à identifier une forme (une caractéristique) définie par un modèle analytique. Par exemple pour le cas d'une ellipse, on cherche à déterminer l'équation

$$h(u,v) = \frac{(u-\alpha)^2}{a} + \frac{(v-\beta)^2}{b} - 1 = 0 \tag{16}$$

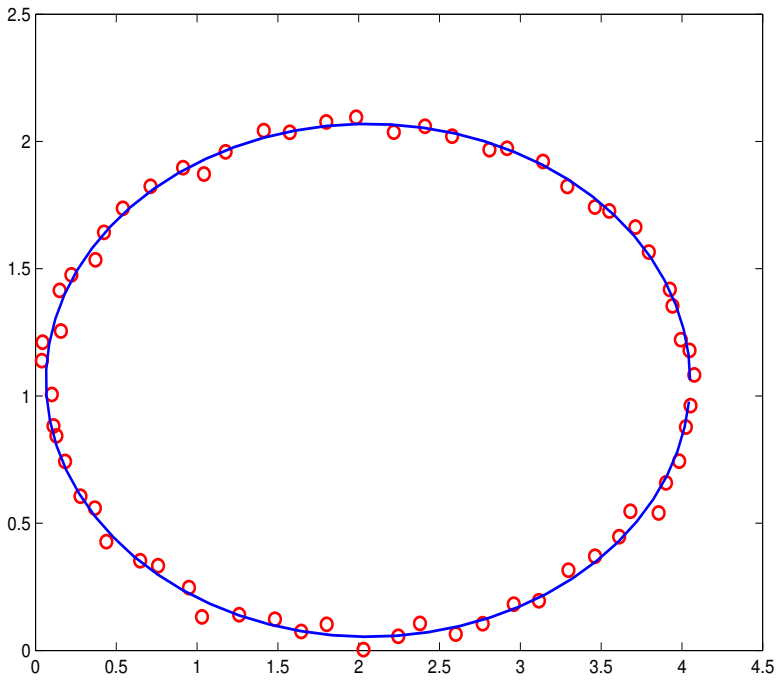
où α, β, a et b sont les paramètres du problème à déterminer. A partir d'un ensemble de mesures défini par un jeu de points $(u_i, v_i),\ i = 1, \cdots, n$, il est possible de formaliser le problème d'identification de la forme comme un problème d'optimisation ayant pour fonction de coût

$$\min_{\alpha,\beta,a,b} \sum_{i=1}^{n} \left(\frac{(u_i-\alpha)^2}{a} + \frac{(v_i-\beta)^2}{b} - 1 \right)^2 \tag{17}$$

Chaque terme de la fonction de coût exprime l'erreur entre la mesure et le modèle.

Introduction

Dans cet exemple, la fonction de coût n'est pas linéaire par rapport aux paramètres α, β, a et b. Il y a plusieurs expressions possibles pour celle-ci. Dans cet exemple, on a fait le choix d'une fonction qui minimise l'erreur d'estimation du modèle et qui est dérivable.



Optimisation dans Matlab : fonctions *lsqcurvefit, lsqnonlin*

Introduction

Quand la fonction $f(x)$ est non convexe et non linéaire, on utilise des algorithmes itératifs de recherche d'extrema. Il en existe un nombre important et ces algorithmes utilisent une approximation de $f(x)$ basée sur un développement limité de Taylor. Ils supposent que $f(x)$ est dérivable et lisse.

- Développement à l'ordre 1 $\Rightarrow$ *méthodes de descente*

$$f(x+h) = f(x) + \nabla f(x)^T h + \Theta(h) \tag{18}$$

- Développement à l'ordre 2 $\Rightarrow$ *méthodes de Newton ou quasi-Newton*

$$f(x+h) = f(x) + \nabla f(x)^T h + \frac{1}{2} h^T \nabla^2 f(x) h + \Theta(h) \tag{19}$$

Méthodes de descente : problématique

Approche : On suppose que l'on connaît un point x et la fonction de coût $f(x)$ en x. On cherche dans le voisinage de x un autre point pour lequel la fonction de coût est plus faible. Une solution simple consiste à réaliser un déplacement en ligne droite. La direction de déplacement est alors appelé *direction de descente*.

Definition 1 : Un vecteur d est une direction de descente s'il existe τ tel que

$$f(x + t\ d) < f(x), \quad t \in [0, \tau] \tag{20}$$

Le principe des méthodes de descente consiste donc à trouver de façon itérative une séquence de points x_k telle que

$$x_{k+1} = x_k + t_k\ d_k \quad \text{avec} \quad f(x_{k+1}) < f(x_k) \tag{21}$$

où d_k et t_k désignent respectivement la direction de descente et le pas de déplacement en x_k. d_k et t_k sont deux inconnues qui doivent être déterminées à chaque itération.

Méthodes de descente : problématique

Considérons le développement limité au 1er ordre de f en $x + t\ d$. On a

$$f(x + t\ d) = f(x) + t\nabla f(x)^T d + t\Theta(t)\ , \quad t > 0 \tag{22}$$

avec $\lim_{t\to 0} \Theta(t) = 0$

Property 1 : Si d est une direction de descente, on a alors

$$\nabla f(x)^T d < 0 \tag{23}$$

Exercise 6 : Démontrer ce résultat.

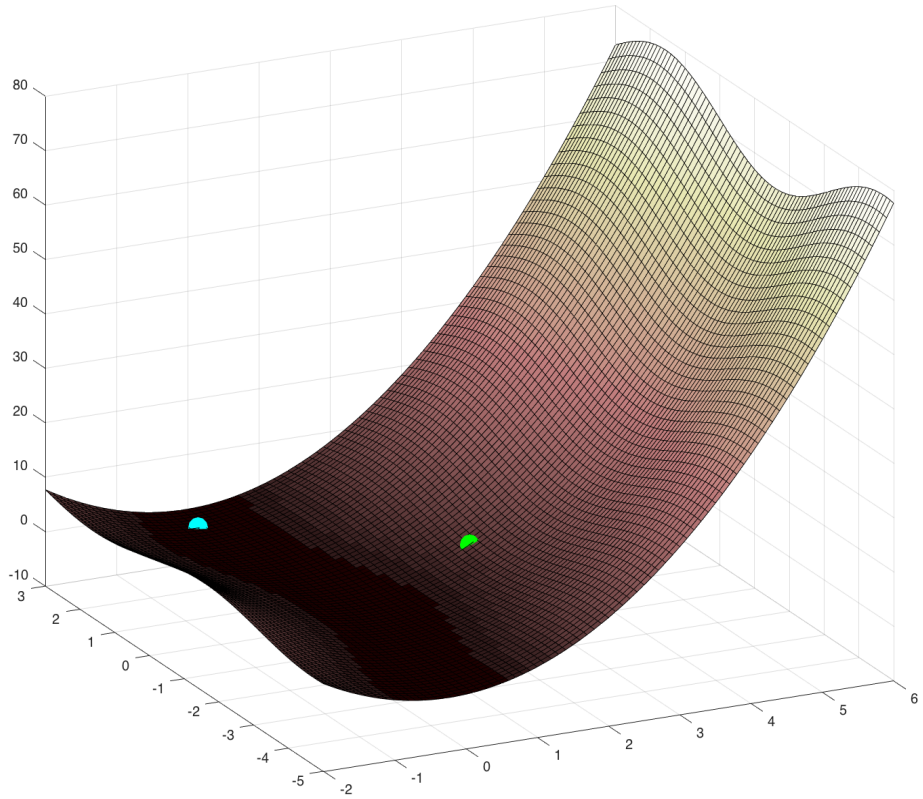
Si on suppose que x_k et d_k sont connus, le pas t_k peut être choisi afin d'obtenir la plus grande diminution possible de f. Dans ce cas, on cherche le pas optimal t_k^* tel que

$$t_k^* = \underset{t}{\text{argmin}}\ \ \Phi(t) \tag{24}$$

où $\Phi(t) = f(x_k + t\ d_k)$ est une fonction monodimensionnelle de t.

Méthodes de descente : problématique

Exercise 7 : On considère la fonction de coût $f(x, y) = 2x^2 + x \sin y$ et le point initial $(x_0 \;\; y_0)^T = (2 \;\; -1)^T$. Démontrer que la direction $d = (-2 \;\; 3)^T$ est une direction de descente. Calculer le nouveau point atteint pour un déplacement de pas optimal t^* selon d.



Méthodes de gradient

Les méthodes de gradient utilisent le gradient de la fonction de coût comme direction de descente. On choisit donc :

$$d_k = -\nabla f(x_k) \tag{25}$$

Ce choix garantit que la grandeur $\nabla f(x_k)^T d_k$ est négative et minimale.

Algorithme simplifié : pas constant $t_k = \rho > 0$

1. Etape 1 : Initialisation $k = 0$ et $x_k = x_0$
2. Etape 2 : Calcul de $d_k = -\nabla f(x_k)$
3. Etape 3 : Calcul de $x_{k+1} = x_k + \rho d_k$
4. Etape 4 : $k + 1 \longrightarrow k$, retour en 2

La convergence de l'algorithme est évaluée en calculant la grandeur $|| x_{k+1} - x_k ||$. Cette grandeur peut être utilisée comme condition d'arrêt : $|| x_{k+1} - x_k || < \epsilon_{Threshold}$. L'algorithme nécessite pour être robuste d'avoir un pas t_k faible et une fonction de coût lisse.

Méthodes de gradient

Algorithme à pas optimal

1. Etape 1 : Initialisation $k = 0$ and $x_k = x_0$
2. Etape 2 : Calcul de $d_k = -\nabla f(x_k)$
3. Etape 3 : Calcul du pas optimal $t_k^* = \underset{t}{\text{argmin}} \ \Phi(t)$ avec $\Phi(t) = f(x_k + t\ d_k)$
4. Etape 4 :Calcul du point $x_{k+1} = x_k + t_k^*\ d_k$
5. Etape 5 : $k + 1 \longrightarrow k$, retour en 2

Property 2 : Deux directions de descente consécutives d_k et d_{k+1} sont orthogonales.

$$d_{k+1}^T d_k = 0 \tag{26}$$

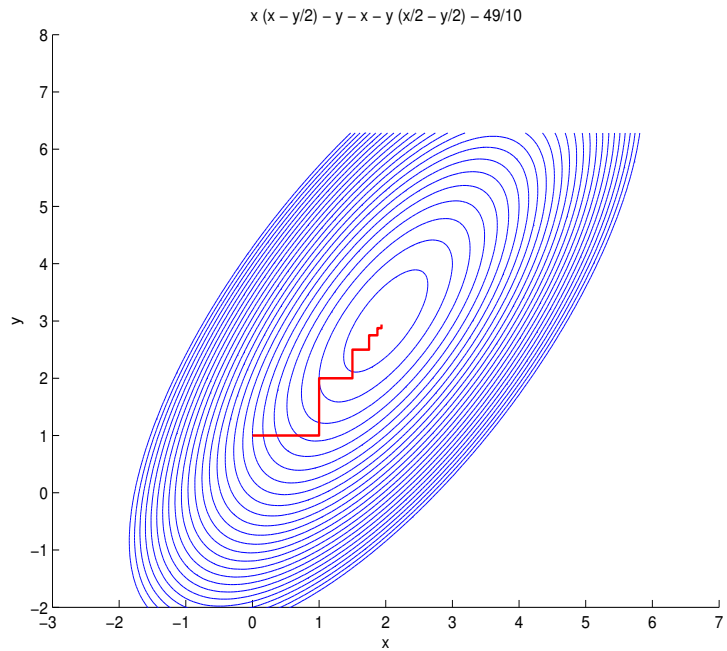
Exercise 8 : Démontrer ce résultat.

Eléments de solution : le pas optimal t_k^* est solution de l'équation : $\Phi'(t) = 0 \Longleftrightarrow \nabla f(x_k + t_k^*\ d_k)^T d_k = 0$ sachant que $x_{k+1} = x_k + t_k^*\ d_k$ et $d_{k+1} = -\nabla f(x_{k+1})$

Méthodes de gradient

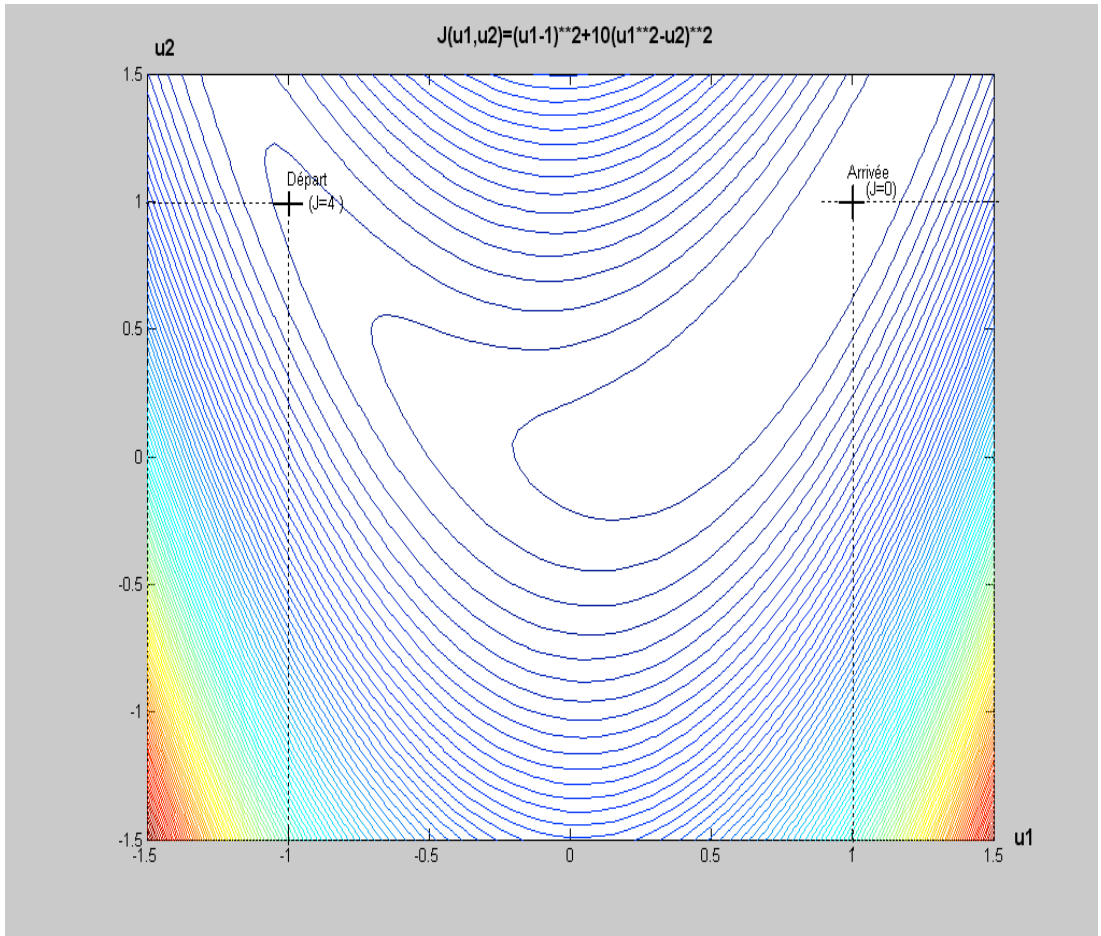
Exercise 9 : On considère la fonction de coût $f(x) = \frac{1}{2}x^T Ax - b^T x$ avec A une matrice symétrique de dimension $n \times n$ et b un vecteur de dimension $n \times 1$.

1. Etablir l'expression analytique du pas optimal (Armijo's relation)
2. Pour $f(x) = x_1^2 + 0.5x_2^2 - x_1x_2 - x_1 - x_2$, calculer les trois premières itérations de l'algorithme de gradient à pas optimal en partant de la condition initiale $(2 \;\; 2)^T$. Comparer le résultat obtenu au résultat théorique.



Méthodes de gradient

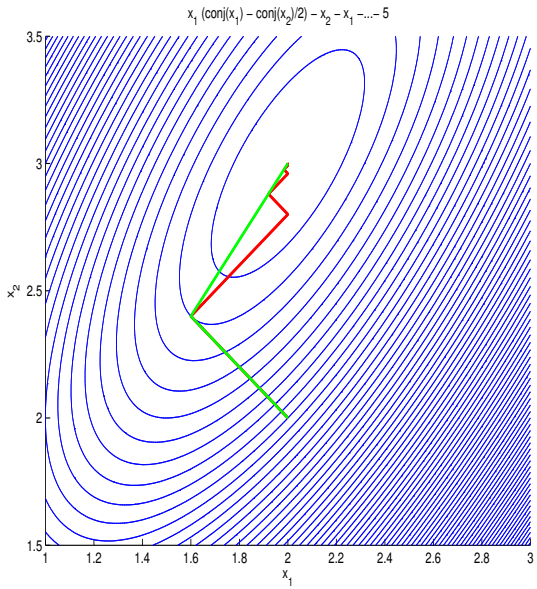
Exercise 10 : Pour la fonction de coût $f(x_1, x_2) = (x_1 - 1)^2 + 10(x_1^2 - x_2)^2$, calculer les deux premières itérations de l'algorithme de gradient à pas optimal pour la condition initiale $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$.

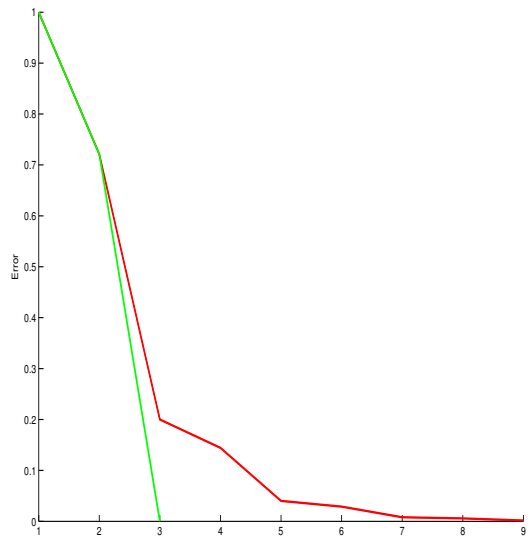


Variantes des méthodes de gradient

Il existe de nombreuses variantes de la méthode de gradient.

- Méthode de gradient conjugué (cf. annexe) : la méthode utilise comme direction de descente les directions conjuguées issues de la matrice Hessienne. Plusieurs algorithmes sont proposés dans la littérature : algorithme de Flechter-Reeves ou de Polak-Ribière





- Méthode de relaxation : les directions de descente coincident avec les directions principales de $\mathbb{R}^n$

Méthode de Newton

La méthode de Newton reposent sur le principe qui consiste à minimiser localement la fonction définie par le développement limité au second ordre de la fonction de coût $f(x)$.
Soit x_0 une solution initiale. Le développement limité de $f(x)$ dans le voisinage de x_0 s'écrit sous la forme :

$$f_0(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0) \quad (27)$$

La méthode consiste à minimiser la fonction $f_0(x)$, ce qui revient à trouver le point x_1 solution de l'équation :

$$\nabla f(x_0) + \nabla^2 f(x_0)(x_1 - x_0) = 0 \quad (28)$$

Il en résulte

$$x_1 = -[\nabla^2 f(x_0)]^{-1} \nabla f(x_0) + x_0 \quad (29)$$

Méthode de Newton

Algorithme simplifié :

1. Initialisation $k = 0$ and $x_k = x_0$
2. Calcul du nouveau point x_{k+1} tel que $x_{k+1} = x_k + \delta_k$ avec δ_k solution de l'équation

$$\nabla^2 f(x_k)\, \delta_k = -\nabla f(x_k) \tag{30}$$

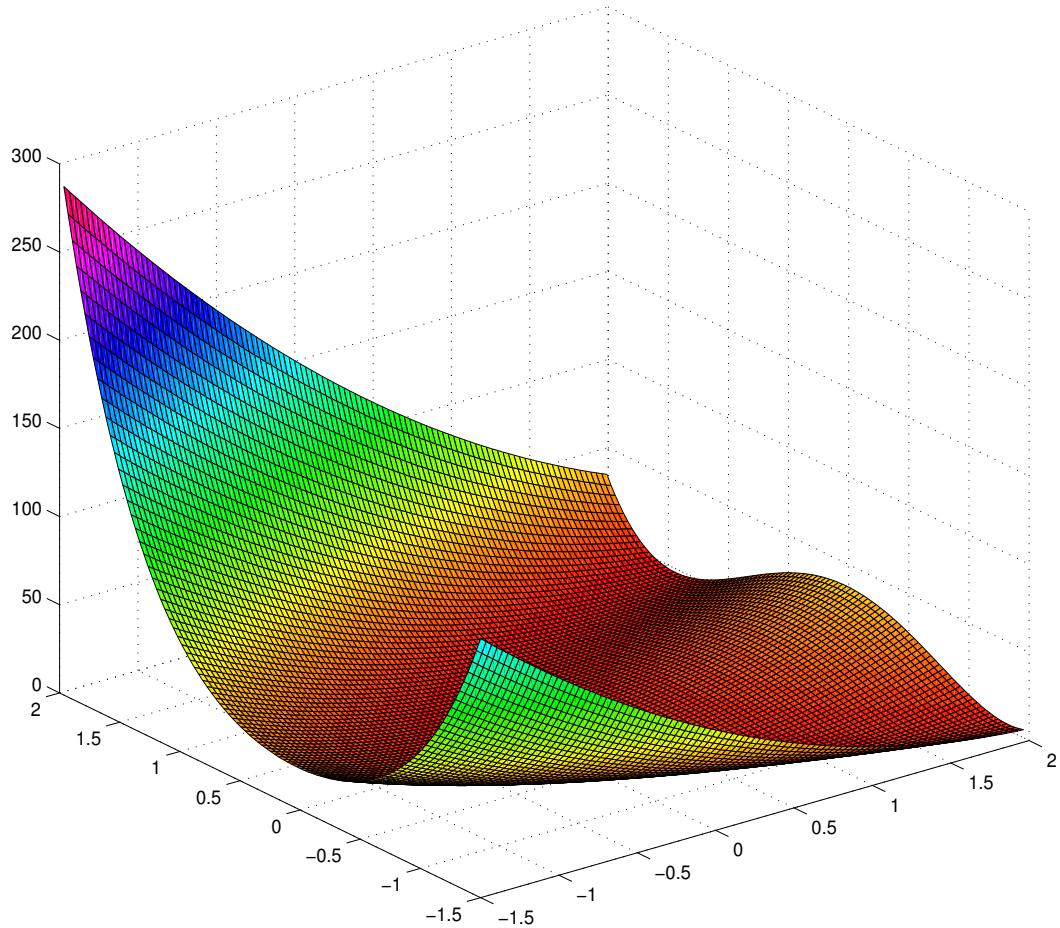
Pour déterminer δ_k il est nécessaire d'inverser la matrice $\nabla^2 f(x_k)$. Plusieurs algorithmes ne calculent pas l'inverse de $\nabla^2 f(x_k)$ mais utilisent une estimation de son inverse : méthodes de quasi-Newton.

Propriétés de l'algorithme

- L'algorithme reste efficace pour des problèmes de taille moyenne (Medium Scale Problem) quand la matrice Hessienne est facile à calculer.
- La direction δ_k n'est pas nécessairement une direction de descente. C'est le cas uniquement si la matrice Hessienne $\nabla^2 f(x_k)$ est définie positive.

Méthode de Newton

Exercise 11 : Calculer les trois premières itérations de l'algorithme de Newton pour la fonction de coût $f(x_1, x_2) = (x_1 - 1)^2 + 10(x_1^2 - x_2)^2$ (fonction de Rosenbrock) à partir du point initial $(-1 \;\; 1)^T$.



Variantes de la méthode de Newton

1. **Méthode de Levenberg-Marquardt :** cette méthode est souvent utilisée afin de résoudre le problème de l'inversion de $\nabla^2 f(x_k)$ et d'obtenir une matrice définie positive. Dans ce cas, la matrice $\nabla^2 f(x_k)$ est remplacée par $\nabla^2 f(x_k) + \mu \mathbb{I}_d$ avec μ un paramètre défini par l'utilisateur. Dans ce cas, δ_k est obtenu en résolvant l'équation

$$[\nabla^2 f(x_k) + \mu \mathbb{I}_d]\, \delta_k = -\nabla f(x_k) \tag{31}$$

2. **Méthodes de quasi-Newton** : les méthodes de quasi-Newton utilisent une approximation de l'inverse de la matrice Hessienne. Le calcul du point suivant est obtenu à l'aide de l'équation suivante

$$x_{k+1} = x_k - \alpha_k S_k \nabla f(x_k) \tag{32}$$

où S_k est une matrice symétrique correspondant à l'approximation de $[\nabla^2 f(x_k)]^{-1}$. L'approche est équivalente à une méthode de descente de pas α_k et de direction $-S_k \nabla f(x_k)$.

Variantes de la méthode de Newton

La matrice S_k est calculée de façon itérative sous la forme $S_{k+1} = S_k + C_k$ en se basant sur la condition de quasi-Newton (cf. annexe). Pour ce calcul plusieurs algorithmes sont proposés dans la littérature :

- Algorithme de Davidon-Fletcher-Powell (DFP)

$$S_{k+1} = S_k + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{S_k \gamma_k \gamma_k^T S_k}{\gamma_k^T S_k \gamma_k} \tag{33}$$

- Algorithme de Broyden-Fletcher-Goldfarb-Shanno (BFGS)

$$S_{k+1} = S_k + \frac{\gamma_k \gamma_k^T}{\gamma_k^T \delta_k} - \frac{S_k \delta_k \delta_k^T S_k}{\delta_k^T S_k \delta_k} \tag{34}$$

avec $\gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ et $\delta_k = x_{k+1} - x_k$

Optimisation non contrainte avec Matlab

- Fonction *fminunc*
  $>>$ x = fminunc(fun,x0,options)
  1. Algorithmes en grande dimension (Large Scale) :
     - Problèmes avec un nombre très important de variables ($>$ 10000) et/ou des fonctions difficiles à résoudre.
     - Initialisation : point intérieur, "trust-region"
     - Utilisation de matrices "sparse".
  2. Algorithmes en moyenne dimension (Medium Scale) :
     - Utilisation du Gradient et Hessienne : fonctions lisses
     - Algorithmes de quasi-Newton : BFGS, DFP
     - Méthode de simplexe (algorithme de Nelder-Mead) pour les fonctions non lisses et discontinues
- Fonction *fminsearch* : utilisation d'un algorithme de simplexe

Section 3 : optimisation non linéaire avec contraintes

- Introduction
- Optimisation avec contraintes d'égalité
- Optimisation avec contraintes d'inégalité

Introduction

L'objectif de cette section est d'étudier les techniques d'optimisation non linéaires en présence de contraintes. Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ la fonction de coût. On considère le problème d'optimisation suivant :

$$\begin{aligned} &\underset{x}{\text{minimize}}\ f(x) \\ &\text{sous les contraintes} \\ &\qquad h_i(x) = 0, \quad i = 1, \cdots, m_{eq} \\ &\qquad g_j(x) \leq 0, \quad j = 1, \cdots, m \\ &\qquad x_k^l \leq x_k \leq x_k^u, \quad k = 1, \cdots, n \end{aligned}$$

où les fonctions $h_i(x)$ et $g_j(x)$ définissent un ensemble de $m_{eq} + m$ contraintes, x_k^l et x_k^u désignent les limites inférieure et supérieure de la variable x_k. La distinction entre égalités et inégalités est importante.

Optimisation avec contraintes d'égalité

Le problème d'optimisation est alors défini par : $\min_x f(x)$ sous les contraintes $h_i(x) = 0,\ i = 1, \cdots, m_{eq}$. Les fonctions $f(x)$ et $h_i(x)$ peuvent être linéaires ou non linéaires.

En fonction de l'expression des contraintes, il est parfois possible par substitution de réduire le nombre de variables en répercutant les contraintes dans la fonction de coût. Dans l'exemple suivant :

$$f(x) = x_1 x_2 x_3$$

sous la contrainte

$$h_1(x) = x_1 + x_2 + x_3 - 1 = 0$$

la contrainte $h_1(x)$ est utilisée afin d'éliminer la variable x_3 et de transformer le problème avec contrainte à 3 variables en un problème sans contrainte à deux variables. On obtient alors :

$$f(x) = x_1 x_2 (1 - x_1 - x_2)$$

Optimisation avec contraintes d'égalité

Méthode des multiplicateurs de Lagrange :
La méthode des multiplicateurs de Lagrange permet de transformer un problème d'optimisation avec contraintes d'égalité en un problème d'optimisation sans contrainte. Pour cela, on définit une nouvelle fonction de coût, appelée fonction de Lagrange, qui s'exprime à partir de la fonction de coût initiale et des contraintes pondérées par un vecteur de coefficients λ, appelé multiplicateurs de Lagrange. La dimension de λ est égale au nombre de contraintes d'égalité. La fonction de Lagrange s'écrit alors sous la forme :

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m_{eq}} \lambda_i h_i(x) = f(x) + \lambda^T h(x) \tag{35}$$

avec $h(x) = (h_1(x), \cdots, h_{m_{eq}}(x))^T$ le vecteur composé des contraintes d'égalité.
La résolution du problème d'optimisation consiste alors à déterminer l'ensemble des extrema de la fonction $L(x, \lambda)$ et à ne retenir que les solutions correspondantes à l'objectif (minimum ou maximum) recherché pour la fonction $f(x)$.

Optimisation avec contraintes d'égalité

Les extrema de $L(x, \lambda)$ sont obtenus en résolvant le système d'équations suivant :

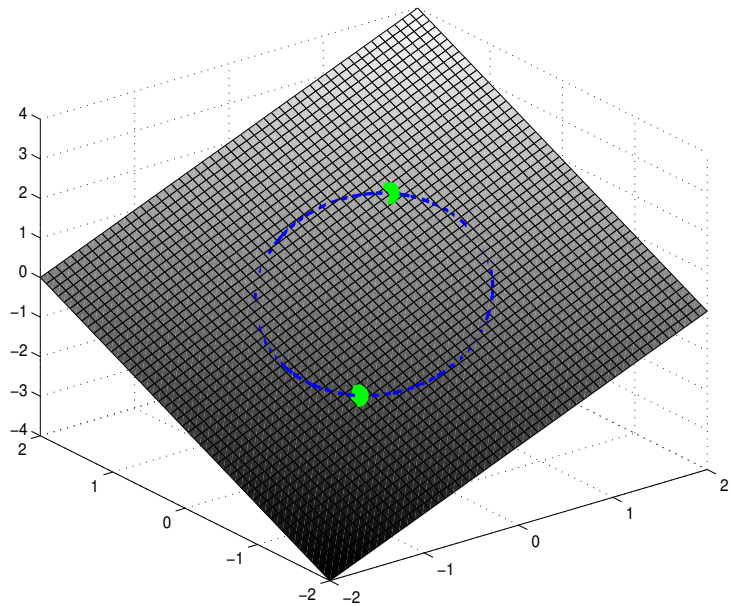
$$\begin{aligned} \nabla_x L(x, \lambda) = \nabla f(x) + \sum_{i=1}^{m_{eq}} \lambda_i \nabla h_i(x) = & \quad 0 \\ \nabla_\lambda L(x, \lambda) = h(x) = & \quad 0 \end{aligned} \tag{36}$$

La résolution du système (36) est équivalent à la résolution de $n + m_{eq}$ équations à $n + m_{eq}$ inconnues. Une solution (x^*, λ^*) de (36) est appelée point stationnaire de la fonction de Lagrange. Il en résulte

$$\begin{aligned} h(x^*) &= 0 \\ L(x^*, \lambda^*) &= f(x^*) \end{aligned}$$

Optimisation avec contraintes d'égalité

Exercise 12 : Résoudre le problème d'optimisation $\min_x (x_1 + x_2)$ sous la contrainte $h(x) = x_1^2 + x_2^2 = 1$. Déterminer les points stationnaires.



Optimisation avec contraintes d'égalité

Exercise 13 : Un fabricant de composants dispose d'un budget publicitaire égal à 500k€. Une étude de marché a permis d'établir une estimation des profits attendus par la fonction suivante :

$$f(x, y) = -x^2 - y^2 + 500x + 1000y$$

où x et y sont les montants (exprimés en k€) investis respectivement dans la publicité par voie de presse et par Internet. Déterminer le montant à investir dans chaque support publicitaire afin de maximiser les profits.

Optimisation avec contraintes d'inégalité

Méthode de Karush-Kuhn-Tucker (KKT)
La méthode des multiplicateurs de Lagrange a été généralisée par Karush-Kuhn-Tucker ([2]) afin de résoudre les problèmes d'optimisation avec contraintes d'égalité et d'inégalité définis par

$$\underset{x}{\text{minimize}}\ f(x)$$

sous les contraintes

$$h_i(x) = 0, \quad i = 1, \cdots, m_{eq}$$
$$g_j(x) \leq 0, \quad j = 1, \cdots, m$$

Pour résoudre ce problème, on considère la fonction de Lagrange suivante :

$$L(x, \lambda, \mu) = f(x) + \lambda^T\ h(x) + \mu^T\ g(x) \tag{37}$$

où $\lambda \in \mathbb{R}^{m_{eq}}$, $\mu \in \mathbb{R}^m, \mu \geq 0$, sont les multiplicateurs de Lagrange , $h(x)$ et $g(x)$ sont les vecteurs de contraintes d'égalité et d'inégalité, respectivement.

2. Kuhn, H. W. ; Tucker, A. W. (1951). "Nonlinear programming".

Optimisation avec contraintes d'inégalité

Conditions de KKT

Si x^* est une solution du problème, alors il existe deux vecteurs λ^* et $\mu^* \geq 0$ tels que :

$$\begin{aligned} \nabla_x L(x^*, \lambda^*, \mu^*) &= 0 \\ h_i(x^*) &= 0, \quad i = 1, \cdots, m_{eq} \\ g_j(x^*) &\leq 0, \quad j = 1, \cdots, m \\ \mu_j^* \; g_j(x^*) &= 0, \quad j = 1, \cdots, m \end{aligned}$$

Les équations $\mu_j^* \; g_j(x^*) = 0$, $j = 1, \cdots, m$ sont appelées conditions d'orthogonalité. Il en résulte :

$$L(x^*, \lambda^*, \mu^*) = f(x^*)$$

Optimisation avec contraintes d'inégalité

Exercise 14 : Soit $f(x) = x_1^2 - x_2$ la fonction de coût. Résoudre le problème d'optimisation $\min_x f(x)$ sous les contraintes : $x_1 + x_2 = 6$, $x_1 - 1 \geq 0$ et $x_1^2 + x_2^2 \leq 26$.

References

- S. Boyd, L.Vandenberghe, *Convex Optimization*, Cambridge
- J.S. Arora, *Introduction to optimal design*, Elsevier
- R. Robere, *Interior Point Methods and Linear Programming*, 2012