

MDI 210 - Optimisation

1 Problèmes de l'analyse numérique

On considère un système linéaire écrit sous la forme matricielle $Ax = b$.

Deux types d'erreurs peuvent être commises :

- Les erreurs d'arrondi, dues aux limites du codage employé.
- Les erreurs de troncature, due à la limite choisie en nombre d'itérations.

Conditionnement d'un système linéaire

Def. On appelle **conditionnement** de A (relativement à la norme $\|\cdot\|$, la quantité $\|A\| \cdot \|A^{-1}\|$, que l'on note $\text{cond}_{\|\cdot\|}(A)$ ou $\text{cond}(A)$.

Dans le cas $A(x + \delta x) = b + \delta b$ on a alors $\frac{\|\delta x\|}{\|x\|} \leq \text{cond}(A) \frac{\|\delta b\|}{\|b\|}$. Dans le cas $(A + \delta A)(x + \delta x) = b$ alors $\frac{\|\delta x\|}{\|x + \delta x\|} \leq \text{cond}(A) \frac{\|\delta A\|}{\|A\|}$.

Une matrice est d'autant mieux conditionnée que son conditionnement est proche de 1.

Th. Soit A une matrice inversible. On a alors :

- $\text{cond}(A) \geq 1$
- $\text{cond}(A) = \text{cond}(A^{-1})$
- $\forall \alpha \neq 0, \text{cond}(\alpha A) = \text{cond}(A)$
- En notant cond_2 le conditionnement associé à $\|\cdot\|_2$ et en notant respectivement $\mu_1(A)$ et $\mu_n(A)$ la plus petite et la plus grande des valeurs singulières de A , $\text{cond}_2(A) = \frac{\mu_n(A)}{\mu_1(A)}$.
- Si A est normale (i.e. $AA^* = A^*A$), $\text{cond}_2(A) = \frac{\max_i |\lambda_i(A)|}{\min_i |\lambda_i(A)|}$ où les $\lambda_i(A)$ représentent les valeurs propres de A .
- Si A est unitaire ou orthogonale, $\text{cond}_2(A) = 1$.
- $\text{cond}_2(A)$ est invariant par transformation unitaire ou orthogonale : si $UU^* = I$ (resp. $UU^T = I$) alors

$$\text{cond}_2(A) = \text{cond}_2(AU) = \text{cond}_2(UA) = \text{cond}_2(U^*AU) \quad (\text{resp. } \text{cond}_2(U^T AU)).$$

Def. Le problème de l'équilibrage d'une matrice consiste à chercher deux matrices D_1 et D_2 diagonales inversibles telle que $\text{cond}(D_1 A D_2) = \inf_{\Delta_1, \Delta_2 \text{ diagonales inversibles}} \text{cond}(\Delta_1 A \Delta_2)$.

On résout alors $Ax = b$ en deux étapes : résolution de $D_1 A D_2 y = D_1 b$ puis de $x = D_2 y$.

En pratique on essaie plus simplement de minimiser le rapport entre le plus grand et le plus petit élément non nul de $A' = \Delta_1 A \Delta_2$. Posons $E = \{(i, j) \in \llbracket 1; n \rrbracket^2 \mid a'_{i,j} \neq 0\}$. On minimise $\frac{\max_{(i,j) \in E} |a'_{i,j}|}{\min_{(i,j) \in E} |a'_{i,j}|}$.

Conditionnement d'un problème de recherche de valeurs propres

Th. Soit A diagonalisable et P une matrice telle que $P^{-1}AP = \text{diag}(\lambda_i)$. Soit $\|\cdot\|$ une norme matricielle telle que, pour toute matrice diagonale $\text{diag}(\alpha_i) : \|\text{diag}(\alpha_i)\| = \max_i |\alpha_i|$. Alors, pour toute matrice δA , $\text{Sp}(A + \delta A) \subset \bigcup_{i=1}^n D_i$ avec $D_i = \{z \in \mathbb{C} \mid |z - \lambda_i| \leq \text{cond}_{\|\cdot\|}(P) \cdot \|\delta A\|\}$.

Def. Le conditionnement $\Gamma(A)$ relatif à la recherche des valeurs propres est le minimum de $\text{cond}_{\|\cdot\|}(P)$ pour P tel que $P^{-1}AP$ soit diagonale.

On a alors $\text{Sp}(A + \delta A) \subset \bigcup_{i=1}^n B(\lambda_i, \Gamma(A) \cdot \|\delta A\|)$.

Th. Soit A symétrique et $B = A + \delta A$ où la perturbation δA est également symétrique. Soit $\alpha_1 \leq \dots \leq \alpha_n$ les valeurs propres de A et $\beta_1 \leq \dots \leq \beta_n$ celles de B . Alors $\forall i \in \llbracket 1; n \rrbracket, |\alpha_i - \beta_i| \leq \|\delta A\|_2$.

2 Résolution de systèmes linéaires

Problème : résoudre $Ax = b$ sachant A inversible.

Méthode de Gauss (pour une matrice quelconque)

Par combinaisons linéaires successives entre lignes de A on se ramène à $(MA)x = Mb$ où MA est triangulaire supérieure. On résout ensuite cette équation par une méthode de remontée.

Algorithme 1 : Étape d'élimination

Entrées : Matrice A .

$A' \leftarrow A$;

tant que $\dim(A') > 1$ **faire**

 pivot $\leftarrow a'_{i,1} \neq 0$ (qui existe car A' est inversible) ;

si $i \neq 1$ **alors**

$L_i \leftrightarrow L_1$;

pour tous les $i > 1$ **tel que** $a_{i,1} \neq 0$ **faire**

$L_i \leftarrow L_i - \frac{a_{i,1}}{a_{1,1}} L_1$;

 // on annule tous les termes de la colonne du pivot situés sous la diagonale

 // notre matrice A' n'a alors que des 0 sous le premier terme, qui est non nul

 // on met alors de côté la première ligne et la première colonne de A'

$A' \leftarrow (a'_{i,j})_{2 \leq i,j}$

On remplace les lignes et colonnes supprimées au fur et à mesure pour obtenir une matrice triangulaire (MA).

Un pivot trop petit en valeur absolue peut causer des erreurs d'arrondi du fait de la division par le pivot. D'où deux stratégies :

- *pivot partiel* : on prend le terme de plus grande valeur absolue de la colonne courante, sur ou en dessous de la diagonale,
- *pivot total* : on choisit le terme de plus grande valeur absolue de la matrice résiduelle, i.e. si on est à l'étape $n - k + 1$, la matrice constituée des k dernières lignes et colonnes (plus coûteux).

La complexité est en $O\left(\frac{2n^3}{3}\right)$ sans choix du pivot.

Méthode de Gauss-Jordan (variante de la précédente)

Dans la phase d'éliminations on élimine également les termes au-dessus de la diagonale. On obtient ainsi une matrice diagonale, ce qui permet de calculer efficacement l'inverse.

Factorisation LU

Th. Soit $A = (a_{i,j}) \in \mathcal{GL}_n(\mathbf{K})$ telle que $\forall k \in \llbracket 1; n \rrbracket, \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix} \in \mathcal{GL}_k(\mathbf{K})$. Alors A admet une factorisation sous la forme $A = LU$ avec L triangulaire inférieure et U triangulaire supérieure. De plus on peut choisir $\forall i \in \llbracket 1; n \rrbracket, (L)_{ii} = 1$ et la décomposition est alors unique.

Cela signifie que, dans l'algorithme de Gauss, les pivots successifs peuvent toujours être pris sur la diagonale. Si la factorisation échoue ou peut toujours permuter des lignes de A pour arriver à une matrice A' qui admet une factorisation LU.

Cette factorisation est utile lorsque plusieurs systèmes linéaires sont à résoudre : on résout un système sur L puis un système sur U .

Méthode de Cholesky (matrices symétriques définies positives)

Th. Soit $A \in \mathcal{S}_n^{++}(\mathbf{K})$. Il existe une matrice triangulaire B vérifiant $A = BB^T$. De plus on peut imposer que tous les éléments diagonaux de B soient tous strictement positifs et la factorisation $A = BB^T$ est alors unique.

En pratique, on calcule B colonne par colonne à partir de $\forall 1 \leq i \leq j \leq n, a_{ij} = \sum_{k=1}^i b_{ik} b_{jk} = a_{ji}$.

Rem. Le déterminant de la matrice peut alors se calculer facilement : $\det(A) = (b_{11} b_{22} \dots b_{nn})^2$.

Un système $Ax = b$ devient alors $BB^T x = b$. Pour résoudre le système on résout $By = b$ puis $B^T x = y$. La complexité de la factorisation suivie de la résolution est en $O\left(\frac{n^3}{3}\right)$.

3 Valeurs propres et vecteurs propres : méthode de Jacobi

Def. Soit le polynôme $P(\lambda) = \lambda^n + a_1\lambda^{n-1} + \dots + a_{n-1}\lambda + a_n$. Est dite « compagne du polynôme P » la matrice suivante :

$$C(P) = \begin{pmatrix} -a_1 & -a_2 & -a_3 & \dots & \dots & -a_{n-1} & -a_n \\ 1 & 0 & & & & & \\ 0 & 1 & 0 & & & & \\ & 0 & 1 & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 0 & 1 & 0 & \\ & & & & 0 & 1 & 0 \end{pmatrix}$$

Prop. Le polynôme caractéristique de $C(P)$ vaut $(-1)^n P(\lambda)$. La matrice a donc pour valeurs propres les racines de P .

Ce lien prouve, par le théorème d'Abel, que la recherche des valeurs propres d'une matrice ne peut se faire en un nombre fini d'opérations au-delà de la dimension 5.

Méthode de Jacobi Soit $A \in \mathcal{S}_n^+(\mathbf{R})$ non diagonale. On dispose de p et q , $p < q$ tels que $a_{pq} \neq 0$. On définit la matrice suivante :

$$\Omega = I_n + \sin(\theta) \cdot (E_{pq} - E_{qp}) + (1 - \cos(\theta)) \cdot (E_{pp} + E_{qq}) .$$

C'est la matrice de rotation d'angle $-\theta$ dans le plan défini par les p^e et q^e vecteurs de la base (donc orthogonale). On pose $B = \Omega^T A \Omega \in \mathcal{S}_n(\mathbf{R})$, $c = \cos(\theta)$, $s = \sin(\theta)$ et $t = \tan(\theta)$. On a alors

$$\begin{cases} b_{ij} = b_{ji} = a_{ij} & \text{si } i \notin \{p, q\} \text{ et } j \notin \{p, q\} \\ b_{pi} = b_{ip} = c \cdot a_{pi} - s \cdot a_{qi} & \text{si } i \notin \{p, q\} \\ b_{qi} = b_{iq} = s \cdot a_{pi} + c \cdot a_{qi} & \text{si } i \notin \{p, q\} \\ b_{pp} = a_{pp} - t \cdot a_{pq} \\ b_{qq} = a_{qq} + t \cdot a_{pq} \end{cases} .$$

Th. Soit $A \in \mathcal{S}_n^+(\mathbf{R})$ et B la matrice obtenue par la construction précédente. On a alors les relations

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2 = \sum_{i=1}^n \sum_{j=1}^n b_{ij}^2 \quad \sum_{i=1}^n a_{ii}^2 + 2a_{pq}^2 = \sum_{i=1}^n b_{ii}^2$$

(le poids de la matrice se déporte sur la diagonale).

Th. La suite des matrices obtenues par la méthode de Jacobi est convergente et converge vers une matrice diagonale contenant les valeurs propres de A .

Pour accélérer la convergence il vaut mieux prendre $|a_{pq}|$ maximum.

Th. Si toutes les valeurs propres de A sont distinctes, alors la suite des produits des matrices Ω converge vers une matrice orthogonale dont les vecteurs colonnes constituent un ensemble orthonormal de vecteurs propres de A .

4 Programmation linéaire : l'algorithme du simplexe

La forme standard d'un problème d'optimisation est la suivante :

- on maximise une forme linéaire $z = \sum_{j=1}^n c_j x_j$, la **fonction objectif**,
- avec m contraintes linéaires : $\forall i \in \llbracket 1 ; m \rrbracket, \sum_{j=1}^n a_{ij} x_j \leq b_i$,
- et n contraintes de positivité : $\forall j \in \llbracket 1 ; n \rrbracket, x_j \geq 0$.

Ces équations déterminent un **polyèdre des contraintes** qui est convexe. Les n -uplets (x_1, \dots, x_n) qui satisfont ces contraintes s'appellent **solutions réalisables** du problème. Ce sont les points intérieurs (au sens large) du polyèdre des contraintes. La **solution optimale** est celle qui maximise z .

S'il n'existe aucune solution réalisable, le problème est dit infaisable. S'il existe des solutions réalisables mais que z n'y admet pas de maximum fini alors le problème est dit non-borné.

Th. Soit un problème de programmation linéaire dont le polyèdre des contraintes est non vide et dont la fonction à maximiser est majorée sur ce polyèdre. Alors le problème admet un maximum fini atteint en au moins un sommet du polyèdre est contraintes.

Pour trouver le maximum on passe alors itérativement d'un sommet à un sommet adjacent de façon à augmenter la valeur de z .

À partir des données initiales on construit un dictionnaire de variables x_1, \dots, x_{n+m} où x_1, \dots, x_n sont appelées *variables de décision*, *variables de choix* ou encore *variables principales* ou *initiales* et x_{n+1}, \dots, x_{n+m} (aussi positives) s'appellent *variables d'écart*, telles que $x_{n+i} = b_i - \sum_{j=1}^n a_{ij}x_j$.

Un dictionnaire est un système d'équations linéaires liant x_1, \dots, x_{n+m} et satisfaisant :

- les équations expriment z et m des $n+m$ variables (**variables de base**) en fonction des n autres variables (**variables hors-base**),
- équivalence avec le dictionnaire définissant les variables d'écart et la fonction objectif.

Pour une base fixée on obtient une **solution basique** en mettant à 0 toutes les variables hors base.

Une solution est dite **dégénérée** lorsque des variables principales sont nulles.

Algorithme 2 : Itération de l'algorithme du simplexe

Entrées : Un dictionnaire :

$$I \uplus J = \llbracket 1; n+m \rrbracket, z = z^* + \sum_{j \in J} c_j x_j, \forall i \in I, x_i = b_i + \sum_{j \in J} a_{ij} x_j, b_i \geq 0$$

si $\forall j \in J, c_j \leq 0$ **alors**

└ Fin de l'algorithme, on retourne z^* qui est une solution optimale.

sinon

└ $j_0 \leftarrow \arg \max_j \{c_j, j \in J\}$ (choix autre que le maximum possible tant que $c_{j_0} > 0$) ;

└ // x_{j_0} est la variable qui va rentrer en base

└ $i_0 \leftarrow \arg \min_i \left\{ \frac{-b_i}{a_{i,j_0}}, i \in I, | a_{i,j_0} < 0 \right\}$;

└ // x_{i_0} est la variable sortante

└ On extrait x_{j_0} de l'expression courante de x_{i_0} ;

└ On remplace x_{j_0} par sa nouvelle expression dans z et dans l'expression des autres x_i ;

└ On réitère avec le nouveau dictionnaire construit ;

Th (Théorème de Bland). Il ne peut y avoir cyclage lorsque, à toute itération effectuée à partir d'un dictionnaire dégénéré, on choisit les variables entrante et sortante comme celles de plus petit indice parmi les candidats possibles.

On obtient le **problème auxiliaire** en ajoutant une variable x_0 positive telle que

$$\forall i \in \llbracket 1; m \rrbracket, \sum_{j=1}^n a_{ij} x_j - x_0 \leq b_i.$$

Ce problème toujours réalisable en prenant x_0 assez grand et s'il n'était pas réalisable initialement (i.e. avec $x_0 = 0$) on peut chercher le x_0 minimum.

Méthode à deux phases : pour un dictionnaire non réalisable a priori on introduit x_0 et une fonction cible $w = -x_0$. On fait rentrer x_0 en base et on itère pour essayer de trouver un dictionnaire réalisable.

5 Dualité en programmation linéaire

S'il existe m réels $y_i \geq 0$ tels que $\forall j \in \llbracket 1; n \rrbracket, \sum_{i=1}^m a_{ij} y_i \geq c_j$ alors on a, pour toute solution réalisable de (P) : $z \leq \sum_{i=1}^m b_i y_i$. Le **problème dual** (D) du problème primal (P) s'écrit : minimiser $\sum_{i=1}^m b_i y_i$ avec les contraintes $\forall j \in \llbracket 1; n \rrbracket, \sum_{i=1}^m a_{ij} y_i \geq c_j$ et $\forall i \in \llbracket 1; m \rrbracket, y_i \geq 0$.

Rem. Le problème dual de (D) est (P) .

Prop. Soit (x_1^*, \dots, x_n^*) une solution réalisable de (P) et (y_1^*, \dots, y_m^*) une solution réalisable de (D) . On a $\sum_{j=1}^n c_j x_j^* \leq \sum_{i=1}^m b_i y_i^*$. En cas d'égalité les deux solutions sont optimales pour leurs problèmes respectifs.

Cor. Si (P) admet une solution réalisable et est non bornée alors (D) est infaisable.

Th (de la dualité). Si (P) a une solution optimale (x_1^*, \dots, x_n^*) alors (D) a une solution optimale (y_1^*, \dots, y_m^*) et $\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*$.

Prop. Si (P) admet une solution primale telle que, dans son dernier dictionnaire obtenu par la méthode du simplexe, $z = z^* + \sum_{k=1}^{n+m} d_k x_k$, alors une solution optimale de (D) est donnée par $\forall i, y_i^* = -d_{n+i}$.

Th (des écarts complémentaires). Une solution (x_1^*, \dots, x_n^*) de (P) est optimale si et seulement s'il existe (y_1^*, \dots, y_m^*) une solution de (D) vérifiant :

- $\forall i \in \llbracket 1 ; m \rrbracket, \left(\sum_{j=1}^n a_{ij} x_j^* < b_i \right) \implies (y_i^* = 0)$
- $\forall i \in \llbracket 1 ; n \rrbracket, \left(x_j^* > 0 \right) \implies \left(\sum_{i=1}^m a_{ij} y_i^* = c_j \right)$

De plus (y_1^*, \dots, y_m^*) constitue une solution optimale de (D) .

Signification économique du dual :

- b_i est la quantité totale de la ressource i ,
- a_{ij} est la quantité de la ressource i consommée par la fabrication d'une unité de produit j ,
- x_j est la quantité fabriquée de produit j ,
- c_j est la valeur unitaire du produit j ,
- y_i est le prix implicite, la valeur unitaire de la ressource i .

Th. Supposons que la base optimale de (P) est non dégénérée. Pour des variations δb_i des b_i on considère le problème (P_δ) avec les contraintes linéaires $\forall i \in \llbracket 1 ; m \rrbracket, \sum_{j=1}^n a_{ij} x_j \leq b_i + \delta b_i$. On suppose que les δb_i sont suffisamment faibles pour que la base optimale de (P) soit encore réalisable pour (P_δ) . La variation de z vaut alors $\sum_{i=1}^m \delta b_i y_i^*$ où (y_1^*, \dots, y_m^*) est solution optimale de (D) .

L'utilisation du problème dual permet de résoudre un problème où la solution nulle n'est pas réalisable mais où $\forall j, c_j \leq 0$. Un tel problème est dit **dual-réalisable**.

6 Optimisation non linéaire sans contrainte

Def. Soit E et F des espaces vectoriels normés de dimensions finies et U un ouvert de E . Une fonction $f: U \rightarrow F$ est différentiable en $a \in U$ s'il existe une application linéaire $df(a): E \rightarrow F$ appelée différentielle de f en a , et une fonction $r: U \rightarrow F$ continue et nulle en a telles que

$$\forall x \in U, f(x) = f(a) + df(a) \cdot (x - a) + r(x) \cdot \|x - a\|.$$

Def. On appelle matrice jacobienne de f en x relativement aux bases \mathcal{B} et \mathcal{B}' la matrice $\text{Jac}_{f;\mathcal{B},\mathcal{B}'}(x) = \text{Mat}(df(x); \mathcal{B}, \mathcal{B}')$.

Prenons maintenant $f: \mathbf{R}^n \rightarrow \mathbf{R}$. On cherche à déterminer ses extrema, locaux ou globaux.

Def. Soit $x \in \mathbf{R}^n$ tel que f est dérivable en x . Le **gradient** de f en x est $\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^T$.

Prop. Soit $A \in \mathfrak{M}_n(\mathbf{R})$ et u et v deux fonctions de \mathbf{R}^n . Alors $d(u^T A) = d(u^T)A$ et $\nabla(\langle u | v \rangle) = \langle du | v \rangle + \langle u | dv \rangle$.

Def. Si f admet des dérivées partielles d'ordre en x , on a sa **matrice hessienne** dans \mathcal{B} la base canonique, $\mathcal{H}f(x) = \text{Jac}_{f;\mathcal{B}}(x) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right)_{1 \leq i, j \leq n}$. Si f est de classe \mathcal{C}^2 , la matrice hessienne est symétrique.

Prop. Si f est de classe \mathcal{C}^2 en a , on peut écrire la formule de Taylor d'ordre 2 :

$$f(a + h) = f(a) + \langle \nabla f(a) | h \rangle + \frac{1}{2} \langle \mathcal{H}f(a) \cdot h | h \rangle + o(\|h\|^2).$$

Th (condition nécessaire d'optimalité). Si f admet un minimum local en x^* , alors $\nabla f(x^*) = 0$ et $\mathcal{H}f(x^*)$ est une matrice positive (i.e. $\forall h \in \mathbf{R}^n, h^T \mathcal{H}f(x^*) h \geq 0$).

Th (condition suffisante d'optimalité). Si f vérifie en x^* , $\nabla f(x^*) = 0$ et $\mathcal{H}f(x^*)$ est une matrice définie positive (i.e. $\forall h \in \mathbf{R}^n \setminus \{0\}, h^T \mathcal{H}f(x^*) h > 0$) alors f admet un minimum local en x^* .

Def. Soit $A \in \mathfrak{S}_n(\mathbf{R})$, $b \in \mathbf{R}^n$ et $c \in \mathbf{R}$. Une application $q: \mathbf{R}^n \rightarrow \mathbf{R}$ de la forme $x \mapsto c + \langle x | c \rangle + \frac{1}{2} \langle Ax | x \rangle$ est appelée **fonction quadratique**.

Pour cette application quadratique on a $\nabla q(x) = b + Ax$ et $\mathcal{H}q(x) = A$.

Fonctions convexes

Th. Si f est convexe et admet des dérivées partielles, alors f admet un minimum global en x^* ssi $\nabla f(x^*) = 0$.

Th. Si f est convexe et admet un minimum local en x^* , alors c'est un minimum global.

Th. Si f est de classe C^2 alors les propositions suivantes sont équivalentes :

- (a) f est convexe,
- (b) $\forall a, h \in \mathbf{R}^n, f(a+h) \geq f(a) + \langle \nabla f(a) | h \rangle$ (la surface de \mathbf{R}^{n+1} d'équation $x_{n+1} = f(x)$ est au-dessus de ses hyperplans tangents),
- (c) pour tout $x \in \mathbf{R}^n, \mathcal{H}f(x)$ est positive.

Généralités sur les méthodes d'optimisation sans contrainte

Pour déterminer un point où f atteint un minimum local, on construit une suite (x_i) qui doit converger vers un point x^* vérifiant une condition nécessaire d'optimalité. On appelle **méthode de descente** toute méthode telle que $\forall k, x_{k+1} = x_k + s_k d_k$ où $s_k \in \mathbf{R}_+$ et $d_k \in \mathbf{R}^n$ est une direction qui vérifie $\langle d_k | \nabla f(x_k) \rangle < 0$. On veut ainsi assurer au minimum $f(x_{k+1}) \leq f(x_k)$.

Lorsque la convergence d'un algorithme est établie, on s'intéresse à sa *vitesse de convergence* :

- Si $\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq \alpha < 1$ pour k assez grand, on dit que la convergence est *linéaire* de taux α .
- Si $\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \xrightarrow{k \rightarrow \infty} 0$ on dit que la convergence est *superlinéaire*.
- Si $\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^\gamma}$ est borné avec $\gamma > 1$, on dit que la convergence est superlinéaire d'ordre γ . Dans le cas $\gamma = 2$ elle est dite quadratique.

Méthodes de gradient

Soit $x_k \in \mathbf{R}^n$ tel que $\nabla f(x_k) \neq 0$ et $d \in \mathbf{R}^n$. Posons, pour $s \in \mathbf{R}, g(s) = f(x_k + sd)$. On dit que d est une *direction de descente* si $g'(0) = \langle d | \nabla f(x_k) \rangle < 0$.

La direction de plus grande pente descendante est donnée par $d = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$.

Algorithme 3 : Méthode de gradient

Entrées : Fonction f à minimiser.

On choisit un point de départ x_0 ;

$k \leftarrow 0$;

tant que un test d'arrêt n'est pas vérifié **faire**

$d_k \leftarrow -\nabla f(x_k)$;
 $s_k \leftarrow \arg \min_{s \geq 0} f(x_k + s d_k)$;
 $x_{k+1} \leftarrow x_k + s_k d_k$;
 $k \leftarrow k + 1$;

Sorties : x_k

La condition d'arrêt peut être $k > N, \|\nabla f(x_k)\|_2 < \epsilon$ ou $|f(x_{k+1}) - f(x_k)| \leq \epsilon$. Les directions successives empruntées sont ici orthogonales.

Méthode de la plus forte pente accélérée

Soit $p \in \mathbf{N}$. À partir d'un point x_k on effectue p itérations de la méthode de la plus forte pente ; on obtient un point y_k et on pose $d_k = y_k - x_k$. Le point x_{k+1} est le point où $f(x_k + s d_k)$ atteint un minimum pour $s > 0$.

Méthode de Newton unidimensionnelle

On construit une suite (x_k) telle que : si $f''(x_k) > 0$ (f strictement convexe autour de x_k) on pose $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$ (point où la forme quadratique obtenue par formule de Taylor à l'ordre 2 atteint son minimum), sinon la méthode échoue.

Dichotomie pour une fonction dérivable

Def. On dit qu'une fonction est **unimodale** s'il existe $x^* \in \mathbf{R}$ tel que la fonction est décroissante sur $]-\infty ; x^*]$ et strictement croissante sur $[x^* ; \infty[$.

On suppose f dérivable et unimodale, donc avec un minimum global. On cherche x_{\min} et x_{\max} tels que $f'(x_{\min}) < 0$ et $f'(x_{\max}) > 0$. Puis on pose, à chaque itération, $x = \frac{x_{\min} + x_{\max}}{2}$. Si $f'(x) > 0$ on remplace x_{\max} par x , sinon on remplace x_{\min} par x . On répète jusqu'à vérifier un critère d'arrêt et l'on converge ainsi vers le minimum, pour lequel $f'(x) = 0$.

Interpolation quadratique

On dispose de trois points $x_1 < x_2 < x_3$ tels que $f(x_2) \leq f(x_1)$ et $f(x_2) \leq f(x_3)$. Par interpolation de ces trois points on peut alors approcher f par une fonction quadratique q qui admet un minimum en un point x_4 . On change alors le triplet :

- Si $f(x_4) \leq f(x_2)$,
 - si $x_4 \leq x_2 : (x_1, x_4, x_2)$,
 - sinon (x_2, x_4, x_3) .
- Sinon
 - si $x_4 \leq x_2 : (x_4, x_2, x_3)$,
 - sinon (x_1, x_2, x_4) .

7 Optimisation non linéaire avec contraintes

Qualification des contraintes

On considère des fonctions f, g_i ($1 \leq i \leq m$) et h_j ($1 \leq j \leq p$) de \mathbf{R}^n dans \mathbf{R} de classe \mathcal{C}^1 . Soit (P) le problème consistant à minimiser $f(x)$ avec $x \in \mathbf{R}^n$ et $\begin{cases} \forall i \in \llbracket 1 ; m \rrbracket, g_i(x) \leq 0 \\ \forall j \in \llbracket 1 ; p \rrbracket, h_j(x) = 0 \end{cases}$ (les contraintes).

On note X le domaine réalisable (i.e. les x qui respectent les contraintes). C'est un fermé de \mathbf{R}^n . Si $\forall i \in \llbracket 1 ; m \rrbracket, \forall x \in X, g_i(x) = 0$ on dit que la contrainte g_i est **saturée** en x . On supposera X non vide.

Th. Si X est borné, alors (P) admet au moins une solution.

Def. On dit que f est **coercive** si $f(x) \xrightarrow{\|x\| \rightarrow +\infty} +\infty$.

Th. Si f est coercive, (P) admet au moins une solution.

Def. On dit qu'une direction d est **admissible** en $x_0 \in X$ s'il existe $\phi: \mathbf{R} \rightarrow \mathbf{R}^n$ telle que :

- $\phi(0) = x_0$,
- pour tout $t > 0$ assez petit, $\phi(t) \in X$,
- la dérivée à droite de ϕ en 0 est d .

Not. On note $A(x_0)$ l'ensemble des directions admissibles en x_0 et $I_0(x_0) = \{i \in \llbracket 1 ; m \rrbracket \mid g_i(x_0) = 0\}$.

Prop. Soit $d \in A(x_0)$, alors $\forall i \in I_0(x_0), \langle d \mid \nabla g_i(x_0) \rangle \leq 0$ et $\forall j \in \llbracket 1 ; p \rrbracket, \langle d \mid \nabla h_j(x_0) \rangle = 0$. On note $B(x_0)$ les directions qui vérifient ces deux critères, d'où $A(x_0) \subset B(x_0)$.

Def. On dit que les contraintes sont **qualifiées** en $x_0 \in X$ si toute direction dans $B(x_0)$ est limite d'une suite de directions de $A(x_0)$.

Prop. Les contraintes sont qualifiées en tout point de X si les conditions suivantes sont vérifiées :

- les fonctions g_i sont convexes,
- les fonctions h_j sont affines,
- il existe $\tilde{x} \in X$ tel que $\forall i \in \llbracket 1 ; m \rrbracket, g_i(\tilde{x}) < 0$ et $\forall j \in \llbracket 1 ; p \rrbracket, h_j(\tilde{x}) = 0$.

Prop. Supposons les fonction h_j affines pour $j \in \llbracket 1 ; p \rrbracket$. Si, en $x_0 \in X$, l'ensemble des gradients $\nabla g_i(x_0)$ où $i \in I_0(x_0)$ et $\nabla h_j(x_0)$ où $j \in \llbracket 1 ; p \rrbracket$ sont linéairement indépendants, alors les contraintes sont qualifiées en x_0 .

Th. On suppose que (P) admet un minimum local en x^* où les contraintes sont qualifiées. Alors, si $d \in B(x^*)$, $\langle d \mid \nabla f(x^*) \rangle \geq 0$ (aucune direction n'est de descente).

Condition de Lagrange

On considère (P) sans les contraintes g_i et avec des fonctions h_j de classe \mathcal{C}^1 .

Th. Soit x^* un minimum local du problème. On suppose que les contraintes sont qualifiées en x^* . Alors il existe des réels μ_1, \dots, μ_p tels que $\nabla f(x^*) = \sum_{j=1}^p \mu_j \nabla h_j(x^*)$.

Th. La condition de Lagrange est suffisante lorsque f est convexe dans un ouvert contenant X et que les h_j ($1 \leq j \leq p$) sont affines.

Condition de Karush, Kuhn et Tucker

On reprend le problème (P) initial avec les g_i et les h_j de classe \mathcal{C}^1 .

Th (Condition de Karush, Kuhn et Tucker). *On suppose les contraintes qualifiées en x^* un minimum local du problème. Alors il existe $\lambda_1, \dots, \lambda_m \in \mathbf{R}_+$ et $\mu_1, \dots, \mu_p \in \mathbf{R}$ tels que $\nabla f(x^*) = \sum_{j \in J} \mu_j \nabla h_j(x^*) - \sum_{i \in I_0(x^*)} \lambda_i \nabla g_i(x^*)$.*

Th. *La condition de Karush, Kuhn et Tucker en x^* est suffisante pour avoir un minimum local lorsque, simultanément, f et les g_i pour $i \in I_0(x^*)$ sont convexes dans un voisinage de x^* , et les h_j pour $j \in \llbracket 1; p \rrbracket$ sont affines dans un voisinage de x^* .*

Méthode de descente avec contraintes

On veut minimiser $f(x)$ avec $\forall j \in \llbracket 1; m \rrbracket, g_i(x) \leq 0$. On approche alors un minimum par une suite (x_k) avec la méthode du gradient en prenant les directions de plus grande descente dans $B(x_k)$, i.e. d qui minimise $\langle d \mid \nabla f(x_k) \rangle$ tout en vérifiant $\|d\| = 1$ et $\forall i \in I_0(x_k), \langle d \mid \nabla g_i(x_k) \rangle \leq 0$.

Cas des fonctions convexes

On suppose ici que les g_i sont convexes, les h_j sont affines et f est convexe.

Def. Soit $C \subset \mathbf{R}^n$. On dit que C est **convexe** si $\forall (x, x') \in C^2, [x; x'] \subset C$.

Rem. Le domaine réalisable de (P) est convexe.

Th. *Si f est strictement convexe, (P) admet au plus une solution optimale.*

Th. *Si X est borné et f strictement convexe, (P) admet une unique solution.*

Th. *Si f strictement convexe et coercive, (P) admet une unique solution.*

Th. *On suppose qu'on a un minimum local en x^* où les contraintes sont qualifiées. Alors (P) admet un minimum global en x^* .*

Th. *On suppose que la condition de Karush, Kuhn et Tucker est vérifiée en x^* où les contraintes sont qualifiées. Avec les hypothèses de cette partie, x^* est un minimum global de (P) . De plus, si f est strictement convexe, x^* est l'unique point où (P) atteint le minimum global.*