

Memento de Programmation Non-Linéaire

Emmanuel Rachelson

2008

1 Problème de PNL

On considère des fonctions f, g_i ($i \in [1, q]$) et h_j ($j \in [1, p]$) de \mathbb{R}^n dans \mathbb{R} , continues, différentiables deux fois. Résoudre le problème de Programmation Non-Linéaire (P_Ω) associé à ces fonctions, c'est trouver $\hat{x} \in \mathbb{R}^n$ tel que :

$$f(x) = \min_{x \in \Omega} f(x)$$
$$\Omega = \{x \in \mathbb{R}^n / \forall i \in [1, q], g_i(x) = 0, \forall j \in [1, p], h_j(x) \leq 0\}$$

Ce problème correspond à chercher le minimum de la fonction f dans le domaine défini par les contraintes g_i et h_j . La fonction f est appelée *fonction objectif*, les g_i sont des *contraintes égalité*, les h_j des *contraintes inégalité*. On note g la fonction de \mathbb{R}^n dans \mathbb{R}^q qui à x associe le vecteur des $g_i(x)$. De même, $h(x)$ est le vecteur des $h_j(x)$.

2 Théorèmes

2.1 Optimisation sans contraintes

En l'absence de contraintes, les minima de f peuvent être trouvés via les conditions du premier et second ordre :

Théorème (Condition du premier ordre). \hat{x} solution locale de $(P_{\mathbb{R}^n}) \Rightarrow \frac{\partial f}{\partial x}(\hat{x}) = 0$

Théorème (Condition du second ordre). $\left. \begin{array}{l} \frac{\partial f}{\partial x}(\hat{x}) = 0 \\ \frac{\partial^2 f}{\partial x^2}(\hat{x}) \text{ définie positive} \end{array} \right\} \Leftrightarrow \hat{x} \text{ solution locale de } (P_{\mathbb{R}^n})$

Remarques.

- $\frac{\partial f}{\partial x}(\hat{x})$ est le *gradient*, ou matrice *Jacobienne* de f en \hat{x} .
- $\frac{\partial^2 f}{\partial x^2}(\hat{x})$ est la matrice *Hessienne* ou *Hessien* de f en \hat{x} .
- La condition du premier ordre est une condition nécessaire.
- La condition du second ordre est nécessaire et suffisante.
- Si f strictement convexe sur \mathbb{R}^n , alors la condition du premier ordre est condition suffisante.

2.2 Cas des ensembles ouverts

Théorème. Soit Ω un ouvert de \mathbb{R}^n : \hat{x} solution de $(P_\Omega) \Leftrightarrow \left\{ \begin{array}{l} \hat{x} \in \Omega \\ \hat{x} \text{ solution de } (P_{\mathbb{R}^n}) \end{array} \right.$

2.3 Théorème de Lagrange

On ne considère que des contraintes égalité. On introduit l'opérateur *Lagrangien* :

$$\forall (x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^q, L(x, \lambda) = f(x) + \sum_{i=1}^q \lambda_i g_i(x) = f(x) + \lambda^T g(x)$$

On introduit également la notion de *direction admissible* en x . On dit que $v \in \mathbb{R}^n$ est admissible en x si et seulement si :

$$\exists \phi \in C^1(\mathbb{R}^+, \mathbb{R}^n) / \begin{cases} \phi(0) = \hat{x} \\ \phi'(0) = v \\ \phi(t) \in \Omega \text{ pour } t \text{ petit} \end{cases}$$

On note $\mathcal{D}_{\hat{x}}(\Omega)$ l'ensemble des directions admissibles en \hat{x} .

Remarques.

- Une direction admissible est une direction “dans laquelle on peut se déplacer sans sortir du domaine”.
- Pour des contraintes égalité, $\mathcal{D}_{\hat{x}}(\Omega) = \ker \left[\frac{\partial g}{\partial x}(\hat{x}) \right] = \left\{ v \in \mathbb{R}^n / \frac{\partial g}{\partial x}(\hat{x}) \cdot v = 0 \right\}$.

Théorème (Théorème de Lagrange). *Si la famille des $\left\{ \frac{\partial g_i}{\partial x}(\hat{x}), i \in [1, q] \right\}$ est une famille libre alors :*

$$\exists \hat{\lambda} \in \mathbb{R}^q / \left\{ \begin{array}{l} \frac{\partial L}{\partial x}(\hat{x}, \hat{\lambda}) = 0 \\ \forall v \in \mathcal{D}_{\hat{x}}(\Omega) \setminus \{0\}, v^T \frac{\partial^2 L}{\partial x^2}(\hat{x}, \hat{\lambda}) v > 0 \end{array} \right\} \Leftrightarrow \hat{x} \text{ est solution locale de } (P_{\Omega})$$

Remarques.

- La condition “ $\left\{ \frac{\partial g_i}{\partial x}(\hat{x}), i \in [1, q] \right\}$ forme une famille libre” se nomme *régularité des contraintes égalité* (on dit alors que les contraintes sont régulières).
- Les contraintes sont régulières si et seulement si $\frac{\partial g}{\partial x}(\hat{x})$ est de rang q .
- Comme dans le cas sans contraintes, on peut décomposer le théorème de Lagrange en une condition nécessaire du premier ordre et une condition suffisante du second ordre.
- Les éléments λ_i s'appellent *paramètres de Lagrange*.

2.4 Théorème de Karush-Kuhn-Tucker

On considère maintenant le cas général avec contraintes égalité et inégalité. Le Lagrangien s'écrit :

$$\forall (x, \lambda, \mu) \in \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^p, L(x, \lambda, \mu) = f(x) + \sum_{i=1}^q \lambda_i g_i(x) + \sum_{j=0}^p \mu_j h_j(x) = f(x) + \lambda^T g(x) + \mu^T h(x)$$

Si on a $h_j(\hat{x}) = 0$, alors on dit que la contrainte h_j est *active* ou *saturée* en \hat{x} . On note $\mathcal{A}(\hat{x})$ l'ensemble des indices des contraintes saturées en \hat{x} .

Théorème (Théorème de KKT). *Si la famille des $\left\{ \frac{\partial g_i}{\partial x}(\hat{x}), i \in [1, q] \right\} \cup \left\{ \frac{\partial h_j}{\partial x}(\hat{x}), j \in \mathcal{A}(\hat{x}) \right\}$ est une famille libre alors :*

$$\exists (\hat{\lambda}, \hat{\mu}) \in \mathbb{R}^q \times \mathbb{R}^{+p} / \left\{ \begin{array}{l} \frac{\partial L}{\partial x}(\hat{x}, \hat{\lambda}, \hat{\mu}) = 0 \\ \forall j \in [1, p], \hat{\mu}_j h_j(\hat{x}) = 0 \\ \forall v \in \mathcal{D}_{\hat{x}}(\Omega) \setminus \{0\}, v^T \frac{\partial^2 L}{\partial x^2}(\hat{x}, \hat{\lambda}, \hat{\mu}) v > 0 \end{array} \right\} \Leftrightarrow \hat{x} \text{ est solution locale de } (P_{\Omega})$$

Remarques.

- La condition “ $\left\{\frac{\partial g_i}{\partial x}(\hat{x}), i \in [1, q]\right\} \cup \left\{\frac{\partial h_j}{\partial x}(\hat{x}), j \in \mathcal{A}(\hat{x})\right\}$ forme une famille libre” se nomme *qualification des contraintes*. On dit que les contraintes sont *qualifiées*. Il existe un théorème de qualification plus général qui sort du cadre de ce cours.
- Si les $\{g_i, i \in [1, q]\} \cup \{h_j, j \in \mathcal{A}(\hat{x})\}$ sont affines et distinctes, alors les contraintes sont qualifiées.
- Comme dans le cas sans contraintes, on peut décomposer le théorème de KKT en une condition nécessaire du premier ordre et une condition suffisante du second ordre.
- Attention, les $\hat{\mu}_j$ appartiennent à \mathbb{R}^+ .
- Les éléments μ_j s'appellent *paramètres de Karush-Kuhn-Tucker* (parfois raccourci en *paramètres de Kuhn et Tucker*).
- Un pont facile entre théorème de Lagrange et théorème de Karush-Kuhn-Tucker revient à considérer toute contrainte égalité $g_i(x) = 0$ comme deux contraintes inégalité $g_i(x) \leq 0$ et $g_i(x) \geq 0$. Attention toutefois aux raccourcis trop rapides : en un point qui respecte les contraintes, ces dernières ne sont plus qualifiées puisque la famille des gradients de contraintes contient ∇g_i et $-\nabla g_i$ (la famille des gradients des contraintes n'est pas libre). On peut s'en sortir en n'introduisant qu'une fois la contrainte g_i dans le Lagrangien sans pouvoir décider du signe du μ_i associé et on retombe alors ... sur le théorème de Lagrange.

2.5 Dualité

On se place dans le cas de contraintes inégalité. On introduit le problème primal associé à (P_Ω) comme la recherche de :

$$\inf_{x \in \mathbb{R}^n} \sup_{\mu \geq 0} L(x, \mu)$$

Ainsi que le problème dual comme la recherche de :

$$\sup_{\mu \geq 0} \inf_{x \in \mathbb{R}^n} L(x, \mu)$$

On a alors le théorème de la dualité :

Théorème (Dualité). *On suppose les contraintes qualifiées en \hat{x} et les f et $(h_j)_{j \in [1, p]}$ convexes.*

$$\begin{aligned} \hat{x} \text{ solution locale de } (P_\Omega) &\Leftrightarrow \exists \hat{\mu} \in \mathbb{R}^{+p} / (\hat{x}, \hat{\mu}) \text{ solution du problème primal} \\ &\Leftrightarrow \exists \hat{\mu} \in \mathbb{R}^{+p} / (\hat{x}, \hat{\mu}) \text{ solution du problème dual} \end{aligned}$$

En d'autres termes :

\hat{x} est solution de (P_Ω) avec les paramètres de KKT $\hat{\mu} \Leftrightarrow (\hat{x}, \hat{\mu})$ est un point selle du Lagrangien de (P_Ω) .

Il existe une version du théorème de la dualité qui prend en compte les contraintes égalité. Bien qu'elle ne rajoute pas de difficulté par rapport à celle présentée ci-dessus, on ne l'abordera pas.

3 En pratique

3.1 Résoudre en utilisant le théorème de KKT

On part du problème:

$$(P_\Omega) : \begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ \forall i \in [1, q], g_i(x) = 0 \\ \forall j \in [1, p], h_j(x) \leq 0 \\ \forall k \in [1, l], m_k(x) < 0 \end{cases}$$

1. On néglige les m_k en utilisant le théorème d'optimisation sur les ouverts. Il faudra vérifier *a posteriori* que la solution trouvée est bien dans l'ouvert défini par les m_k .
2. On montre que les contraintes g_i et h_j sont qualifiées.
3. On écrit le Lagrangien.
4. On dérive le Lagrangien et on résout le système de $n + p + q$ équations à $n + p + q$ inconnues :

$$\begin{aligned} \frac{\partial L}{\partial x}(x, \lambda, \mu) &= 0 \\ \forall i \in [1, q], g_i(x) &= 0 \\ \forall j \in [1, p], \mu_j h_j(x) &= 0 \end{aligned}$$

La résolution de ce système est probablement l'étape la plus fastidieuse de la méthode puisqu'elle impose de traiter tous les cas de " $\mu_j = 0$ ou $h_j(x) = 0$ " découlant des équations " $\mu_j h_j(x) = 0$ ". Il est souvent pertinent de chercher une solution pour tous les μ_j nuls (aucune hypothèse sur la saturation des contraintes) et de progresser en les saturant au fur et à mesure mais ce n'est en aucun cas une règle d'or. Bien poser les calculs pour regrouper les cas est souvent utile.

Cette résolution au premier ordre fournit un ensemble de points $\{(\hat{x}, \hat{\lambda}, \hat{\mu})\}$ parmi lesquels figurent nécessairement les optima de la fonction.

5. En chacun des $(\hat{x}, \hat{\lambda}, \hat{\mu})$, on vérifie si $\forall v \in \mathcal{D}_{\hat{x}}(\Omega) \setminus \{0\}, v^T \frac{\partial^2 L}{\partial x^2}(\hat{x}, \hat{\lambda}, \hat{\mu}) v > 0$. Généralement, on essaie de le montrer pour tout v de \mathbb{R}^n , puis, si on n'y arrive pas, on regarde le cas $v \in \mathcal{D}_{\hat{x}}(\Omega) \setminus \{0\}$ ou on cherche à montrer que c'est faux (faire un dessin aide souvent !).
6. On vérifie que les $(\hat{x}, \hat{\lambda}, \hat{\mu})$ restants satisfont les contraintes m_k .
7. Les points $(\hat{x}, \hat{\lambda}, \hat{\mu})$ sont alors des minima locaux de f qui satisfont toutes les contraintes. On peut chercher le minimum global en les comparant entre eux.

Remarques.

- Attention, le fait qu'on cherche des μ_j positifs ou nuls vient du fait qu'on a écrit le problème sous la forme de contraintes inégalité "inférieur ou égal". Si l'on change le sens de ces contraintes, le signe des μ_j change. Afin d'éviter les erreurs d'étourderie ou les confusions, il est préférable de toujours écrire les problèmes de la même façon, *ie.* les contraintes égalité " $= 0$ " et les inégalité " ≤ 0 ".
- De même, le fait qu'on cherche un $\frac{\partial^2 L}{\partial x^2}$ semi-défini (ou défini) positif vient du fait que l'on cherche un minimum (on l'aurait cherché négatif pour un maximum). Pour éviter confusions et étourderies, même chose : toujours écrire les problèmes d'optimisation comme des problèmes de minimisation.
- De nombreux exercices se font avec un nombre de variables réduit en x . Certains problèmes de la vie courante aussi. Dans ces cas, là ... faites un dessin ! Cela permet d'avoir une bonne idée de ce qu'il faut chercher (notamment pour l'étape où l'on vérifie si les candidats à l'optimalité sont des minima ou non). Dans d'autres cas, c'est le nombre de contraintes qui est faible. Pensez alors à utiliser le

théorème de la dualité et à faire de nouveau un dessin pour fixer les idées. Malheureusement, ça ne marche pas toujours (la plupart des problèmes réels non triviaux ont beaucoup de contraintes et beaucoup de variables) mais ce serait dommage de s'en priver quand on peut le faire.

- Parfois, certaines contraintes égalité mènent directement à l'élimination de certaines variables. Ne pas se lancer aveuglément dans les calculs et prendre le temps de considérer le problème sous plusieurs aspects (notamment le sens physique associé !) peut être profitable. Commencer par prendre quelques minutes pour visualiser le domaine Ω et la forme de la fonction f est souvent un bon départ.

3.2 Résoudre en utilisant le théorème de la dualité

On commence avec le problème (ou on s'y ramène) :

$$(P_\Omega) : \begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ \forall j \in [1, p], h_j(x) \leq 0 \end{cases}$$

1. Vérifier que les contraintes sont qualifiées et que f et les h_j sont convexes (au moins localement et restreindre le domaine en conséquence).
2. Ecrire le Lagrangien.
3. Résoudre le problème d'optimisation sans contraintes $\inf_{x \in \mathbb{R}^n} L(x, \mu)$ en dérivant le Lagrangien par rapport à x . On obtient $\hat{x}(\mu)$.
4. Poser la *fonction duale* $\psi(\mu) = L(\hat{x}(\mu), \mu)$.
5. Résoudre le problème $\sup_{\mu \geq 0} \psi(\mu)$ en dérivant ψ par rapport à μ et distinguer les cas selon le signe de μ .

4 Programmation quadratique

Parmi les problèmes de programmation non-linéaire, la famille des problèmes de programmation quadratique est si souvent traitée qu'elle mérite un petit paragraphe. On la retrouve notamment en commande optimale, en optimisation de coûts économiques, en dimensionnement de structures, etc. Un problème de programmation quadratique correspond souvent à la minimisation d'une grandeur homogène à une énergie sous des contraintes de conception délimitant les bornes physiques des paramètres.

Avec $Q \in \mathbb{R}^{n \times n}$, b et $a_j \in \mathbb{R}^n$, k et $c_j \in \mathbb{R}$, on pose $f(x) = \frac{1}{2}x^T Qx + b^T x + k$ et $h_j(x) = a_j^T x + c_j$. Résoudre le problème de minimisation de f sous les contraintes $h_j(x) \leq 0$ est la forme générale du problème de programmation quadratique.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & \frac{1}{2}x^T Qx + b^T x + k \\ & a_j^T x + c_j \leq 0 \end{aligned}$$

La résolution s'effectue facilement en appliquant le théorème de KKT.

Remarques.

- Les contraintes sont immédiatement qualifiées si les couples (a_j, c_j) sont distincts deux à deux.
- Le gradient du Lagrangien en x vaut $Qx + b + \sum_{j=1}^q a_j$
- Le Hessien du Lagrangien vaut (partout) Q .
- La solution du problème sans contraintes vaut $-Q^{-1}b$ si Q est définie positive (donc f convexe et Q inversible).

5 Méthodes numériques

5.1 Optimisation sans contraintes

Deux grandes familles de méthodes :

- Les méthodes du premier ordre qui utilisent uniquement le gradient de f .
- Les méthodes du second ordre qui nécessitent de connaître le Hessien de f .

Autour du minimum \hat{x} , on a :

$$f(x) \simeq f(\hat{x}) + \frac{1}{2} (x - \hat{x})^T \frac{\partial^2 f}{\partial x^2}(\hat{x}) (x - \hat{x})$$

La vitesse de convergence des méthodes numériques peut donc être raisonnablement évaluée sur la base des problèmes de programmation quadratique.

5.1.1 Algorithme du gradient

C'est une méthode du premier ordre qui consiste à faire des "pas" dans la direction de plus forte pente.

Algorithme (Algorithme du gradient).

$$\left| \begin{array}{l} x_0 \text{ donné} \\ \text{Tant que } \nabla f(x_k) \neq 0 \text{ faire :} \\ \quad d_k = -\nabla f(x_k) \\ \quad t_k = \underset{t \in \mathbb{R}}{\operatorname{argmin}} f(x_k + t d_k) \\ \quad x_{k+1} = x_k + t_k d_k \end{array} \right.$$

Théorème (Convergence de la méthode de plus forte pente).

$\left. \begin{array}{l} f \in C^1(\mathbb{R}^n, \mathbb{R}) \\ \lim_{\|x\| \rightarrow \infty} f(x) = \infty \text{ (coercivité)} \\ t_k / \forall t \in \mathbb{R} f(x_k + t d_k) \leq f(x_k + t d_k) \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} x_k \text{ bornée (donc admet des points d'accumulation).} \\ \text{Ces points d'accumulation sont des extremas locaux de } f \end{array} \right.$

Si en plus, f est strictement convexe, alors x_k converge vers l'unique minimum global sur \mathbb{R}^n .

Remarques.

- Deux directions de descente successives d_k et d_{k+1} sont orthogonales.
- Si on note α et β respectivement la plus petite et la plus grande valeur propre de la matrice A , alors en appliquant l'algorithme du gradient au problème de programmation quadratique associé à A et en notant $E_k = (x_k - \hat{x})^T A (x_k - \hat{x})$ le terme d'erreur, on a :

$$E_k \leq E_0 \left(\frac{\beta - \alpha}{\beta + \alpha} \right)^{2k}$$

On en déduit que plus la matrice A est mal conditionnée (plus la fonction a une forme de vallée étirée), plus la convergence est mal bornée en théorie.

- Il est intéressant de noter que cette borne théorique n'est qu'une borne supérieure et que la convergence peut se faire plus rapidement. Pour un problème quadratique en dimension deux extrêmement mal conditionné, le nombre d'itération avant convergence tend vers deux. Saurez-vous expliquer pourquoi ?

5.1.2 Méthode des gradients conjugués

L'objectif de la méthode des gradients conjugués est d'accélérer la convergence des méthodes de gradient en prenant des directions de descente qui dépendent directement du conditionnement de A . On dit que deux directions x et y de \mathbb{R}^n sont conjuguées par rapport à $A \in \mathbb{R}^{n \times n}$ si $x^T A y = 0$.

Algorithme (Méthode des gradients conjugués (Hestenes-Stiefel, 1952)).

$$\left| \begin{array}{l} x_0 \text{ donné} \\ d_0 = -\nabla f(x_0) \\ \text{Itération } k \\ t_k = \frac{-(\nabla f(x_k))^T \nabla f(x_k)}{d_k^T A d_k} \\ x_{k+1} = x_k + t_k d_k \\ d_{k+1} = -\nabla f(x_{k+1}) + \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2} d_k \end{array} \right.$$

Remarques.

- $\forall i \neq j, d_i^T A d_j = 0$: les directions de descentes sont toutes mutuellement conjuguées.
- Pour des fonctions quadratiques en dimension n , l'algorithme converge en au plus n itérations

Dans le cas des fonctions non-quadratiques, deux variantes sont à retenir :

Algorithme (Fletcher et Reeves, 1964). *L'algorithme est le même, t_k est obtenu en minimisant $f(x_k + t d_k)$.*

Algorithme (Polak et Ribière, 1969). *On remplace la famille des d_k par celle des :*

$$d_{k+1} = -\nabla f(x_{k+1}) + \frac{(\nabla f(x_{k+1}))^T (\nabla f(x_{k+1}) - \nabla f(x_k))}{\|\nabla f(x_k)\|^2} d_k$$

Remarques.

- De façon générale, la convergence de la méthode des gradients conjugués est assurée si on réinitialise périodiquement la direction de descente d_k .
- Dans le cas quadratique, les deux variantes se ramènent à l'algorithme général.
- C'est une méthode assez efficace et peu coûteuse : la seule grandeur à évaluer est le gradient de f en x_{k+1} et entre deux itérations on ne stocke que $(x_k, \nabla f(x_k), d_k)$.

5.1.3 Méthode de Newton

Plutôt que de chercher des points qui font décroître la valeur de f on cherche des points qui annulent ∇f . On pose $F(x) = \nabla f(x)$. L'idée est de trouver un chemin qui mène à $F(x) = 0$. On a : $F(x_k + d) = F(x_k) + DF(x_k) \cdot d + o(d)$. Pour faire converger F vers 0 on prend d tel que $F(x_k) + DF(x_k) \cdot d = 0$.

Algorithme (Méthode de Newton).

$$\left| \begin{array}{l} x_0 \text{ donné} \\ \text{Itération } k \\ \text{Si } DF(x_k) \text{ inversible, } d_k = -(DF(x_k))^{-1} F(x_k) \\ x_{k+1} = x_k + d_k \end{array} \right.$$

Remarques.

- La méthode de Newton nécessite que f soit localement strictement convexe pour pouvoir inverser $DF(x_k) = \frac{\partial^2 f}{\partial x^2}(x_k)$.

- On peut prouver localement la convergence de la méthode de Newton :

$$\text{Si } \begin{cases} F \in C^1(\mathbb{R}^n, \mathbb{R}) \text{ (donc } f \in C^2(\mathbb{R}^n, \mathbb{R})) \\ \exists \hat{x} \in \mathbb{R}^n / F(\hat{x}) = 0 \\ DF(\hat{x}) \text{ inversible} \\ DF \text{ } L\text{-Lipschitzienne } (\|DF(x) - DF(x')\| \leq \|x - x'\|) \end{cases}, \text{ alors :}$$

$$\exists r > 0 / \left. \begin{array}{l} \forall x_0 \in \mathcal{B}_r(\hat{x}) \\ x_{k+1} = x_k - (DF(x_k))^{-1} F(x_k) \end{array} \right\} \Rightarrow \begin{cases} x_k \in \mathcal{B}_r(\hat{x}) \\ \lim_{i \rightarrow \infty} x_i = \hat{x} \end{cases}$$

- La vitesse de diminution de l'erreur est quadratique : $\forall k \in \mathbb{N}^* \|x_{k+1} - \hat{x}\| \leq c \|x_k - \hat{x}\|^2$
- Pour une fonction f quadratique, la méthode de Newton converge en une unique itération.

L'inconvénient de la méthode de Newton réside dans le calcul du Hessien $\frac{\partial^2 f}{\partial x^2}(x_k)$. Les méthodes de Newton approchées, ou algorithmes quasi-Newton, remplacent les évaluations successives de $\frac{\partial^2 f}{\partial x^2}(x_k)$ par une suite de matrices définies positives H_k . Deux versions sont à retenir, l'algorithme DFP et l'algorithme BFGS.

Algorithme (DFP : Davidson, Fletcher et Power, 1959-1963).

$$\left| \begin{array}{l} x_0 \text{ donné} \\ H_0 \text{ donné} \\ \text{Itération } k \\ d_k = -H_k F(x_k) \\ \lambda_k \text{ minimise } f(x_k + t d_k) \\ \Delta_k = \lambda_k d_k \\ x_{k+1} = x_k + \Delta_k \\ \sigma_k = F(x_{k+1}) - F(x_k) \\ H_{k+1} = H_k + \frac{\Delta_k \Delta_k^T}{\Delta_k^T \sigma_k} - \frac{H_k \sigma_k \sigma_k^T H_k}{\sigma_k^T H_k \sigma_k} \end{array} \right.$$

Algorithme (BFGS : Broyden, Fletcher, Goldfarb, Shannon, 1969-1970). Comme DFP, mais on prend :

$$H_{k+1} = H_k + \left(1 + \frac{\sigma_k^T H_k \sigma_k}{\sigma_k^T \Delta_k}\right) \frac{\Delta_k \Delta_k^T}{\Delta_k^T \sigma_k} - \frac{1}{\sigma_k^T \Delta_k} (\Delta_k \sigma_k^T H_k + H_k \sigma_k \Delta_k^T)$$

Remarques.

- DFP et BFGS convergent en au plus n itérations pour une fonction quadratique.
- La convergence de DFP et BFGS est garantie si la suite H_k est régulièrement réinitialisée.

5.2 Optimisation sous contraintes

La résolution des problèmes de PNL en pratique fait souvent appel à un peu de flair dans le choix de la modélisation (c'est le premier travail de l'ingénieur) et se base sur les deux mêmes méthodes principales que la résolution à la main. On distingue donc :

- Les méthodes primales, qui raisonnent directement sur x , cherchant à minimiser f en résolvant les équations du théorème de KKT.
- Les méthodes duales, qui cherchent un point selle du Lagrangien et cherchent donc à résoudre le problème dual.

Dans tous les cas, ces méthodes finissent par faire appel aux méthodes d'optimisation sans contrainte présentées plus haut.

5.2.1 Changements de variables

L'idée est de simplifier le problème pour se ramener à un problème plus simple.

Dans le cas de contraintes égalité, il n'y a pas de raison de se priver d'éliminer certaines variables dès que c'est possible. C'est la généralisation du théorème des fonctions implicites :

Si on a $\frac{\partial g}{\partial x} \neq 0$ alors la contrainte $g(x) = 0$ implique qu'il existe un voisinage $U \subset \mathbb{R}^{n-1}$ et une fonction ϕ définie sur U et à valeurs dans \mathbb{R} telle que :

$$\forall (x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n) \in \mathbb{R}^{n-1} \cap U, x_j = \phi(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n)$$

Le problème $\min_{g(x)=0} f(x)$ se ramène donc à :

$$\min_{(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n) \in \mathbb{R}^{n-1} \cap U} f((x_1, \dots, x_{j-1}, \phi(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n), x_{j+1}, \dots, x_n))$$

Si U est assez grand, le problème peut se ramener à un problème sans contraintes.

Par ailleurs le problème $\min_{a \leq x \leq b} f(x)$ est équivalent au problème $\min_{y \in \mathbb{R}} g(y)$ avec :

$$x = a + (b - a)^2 \sin y$$

5.2.2 Méthode des directions réalisables

Cette catégorie d'algorithmes généralise l'algorithme du gradient en calculant à chaque pas l'ensemble des directions admissibles $\mathcal{D}_{x_k}(\Omega)$ et en cherchant une direction de descente dans $\mathcal{D}_{x_k}(\Omega)$.

5.2.3 Méthode des pénalités

L'idée est de résoudre une séquence de problèmes sans contraintes dans lequel on pénalise de plus en plus la solution avec les contraintes.

On pose :

$$h_+(x) = \begin{bmatrix} \max\{0, h_1(x)\} \\ \vdots \\ \max\{0, h_p(x)\} \end{bmatrix}$$

Algorithme (Méthode de pénalisation).

$$\left| \begin{array}{l} F(x) = \|g(x)\|^2 + \|h_+(x)\|^2 \\ (c_i)_{i \in \mathbb{N}} \text{ est une suite qui tend vers l'infini} \\ x_0 = \text{solution de } \min_{x \in \mathbb{R}^n} f(x) \\ \text{Itération } k \\ (P_k) \text{ est le problème } \min_{x \in \mathbb{R}^n} (f(x) + c_k F(x)) \\ x_k = \text{solution de } (P_k) \end{array} \right.$$

Si la suite des x_i converge alors sa limite est une solution de :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} f(x) \\ h(x) \leq 0 \\ g(x) = 0 \end{aligned}$$

5.2.4 Résolution directe des équations de KKT

L'algorithme correspond à utiliser une méthode de Newton pour résoudre le système d'équations :

$$\begin{aligned}\frac{\partial L}{\partial x}(x, \lambda, \mu) &= 0 \\ \forall i \in [1, q], g_i(x) &= 0 \\ \forall j \in [1, p], \mu_j h_j(x) &= 0\end{aligned}$$

5.2.5 Programmation quadratique séquentielle

L'idée est d'approcher en chaque point x_k le problème en linéarisant les contraintes et en développant f au second ordre. On résout ainsi une séquence de problèmes quadratiques simples. Chaque problème s'écrit :

$$(P_k) : \begin{cases} \min_{u \in \mathbb{R}^n} f(x_k) + \frac{\partial f}{\partial x}(x_k)u + \frac{1}{2}u^T \frac{\partial^2 f}{\partial x^2} u \\ \forall i \in [1, q] g_i(x_k) + \frac{\partial g_i}{\partial x}(x_k)u = 0 \\ \forall j \in [1, p] h_j(x_k) + \frac{\partial h_j}{\partial x}(x_k)u \leq 0 \end{cases}$$

Le nouveau point x s'écrit alors $x_{k+1} = x_k + \hat{u}$.