

Présentation du projet

1. Problème d'optimisation
2. Algorithme SQP
3. Logiciel
4. Organisation du projet

1 Problème d'optimisation

- Formulation du problème
- Lagrangien
- Conditions d'optimalité
- Système non linéaire
- Méthode de Newton

1 Problème d'optimisation

Formulation

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0$$

→ problème d'optimisation non linéaire avec contrainte

Notations

- x : n **variables** ou paramètres
ou inconnues → vecteur de \mathbb{R}^n
- f : **critère** ou fonction **coût**
ou fonction objectif → fonction de \mathbb{R}^n dans \mathbb{R}
- c : m **contraintes d'égalité** → fonction de \mathbb{R}^n dans \mathbb{R}^m

Résolution pratique

Pas de solution analytique

→ méthodes numériques itératives

→ recherche d'un **minimum local**

1 Problème d'optimisation

Problème avec contrainte

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \quad \rightarrow \text{ problème noté (PO)}$$

Fonction de Lagrange (ou lagrangien)

- Le **lagrangien** du problème (PO) est la fonction L de \mathbb{R}^{n+m} dans \mathbb{R}

$$L(x, \lambda) = f(x) + \lambda^T c(x) \quad \Leftrightarrow \quad L(x, \lambda) = f(x) + \sum_{j=1}^m \lambda_j c_j(x)$$

- Les λ_j sont les **multiplicateurs de Lagrange** \approx coefficients de pénalisation des contraintes
 - \rightarrow 1 multiplicateur par contrainte
 - \rightarrow expression des conditions d'optimalité à partir du lagrangien

1 Problème d'optimisation

Conditions d'optimalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \quad \rightarrow \text{lagrangien } L(x, \lambda) = f(x) + \lambda^T c(x)$$

Conditions nécessaire d'ordre 1

- x^* minimum local \Rightarrow Il existe un unique $\lambda^* \in \mathbb{R}^m$ tel que : $\nabla L(x^*, \lambda^*) = 0$
 $\rightarrow (x^*, \lambda^*)$ est un point stationnaire du lagrangien

Conditions de Karush-Kuhn-Tucker (**conditions KKT**)

- Il faut résoudre un système d'équations non linéaires de taille $n+m$:

$$\nabla L(x, \lambda) = 0 \Leftrightarrow \begin{cases} \nabla_x L(x, \lambda) = 0 \\ \nabla_\lambda L(x, \lambda) = 0 \end{cases} \Leftrightarrow \begin{cases} \nabla f(x) + \lambda^T \nabla c(x) = 0 \\ c(x) = 0 \end{cases}$$

Projet Optimisation

1 Problème d'optimisation

Systeme non linéaire

$$F(z) = 0 \rightarrow \text{systeme de } N \text{ équations à } N \text{ inconnues} \Leftrightarrow \begin{cases} F_1(z_1, \dots, z_N) = 0 \\ \vdots \\ F_N(z_1, \dots, z_N) = 0 \end{cases}$$

Méthode de Newton

- Valeur initiale z^0 \rightarrow Nouvelle valeur $z^1 = z^0 + \delta z$ telle que $F(z^1) = 0$
- Ordre 1 :
$$F(z^0 + \delta z) \approx F(z^0) + \nabla F(z^0)^T \delta z = 0$$
$$\Rightarrow \delta z = -\nabla F(z^0)^{-T} F(z^0)$$
$$\Rightarrow \boxed{z^1 = z^0 - \nabla F(z^0)^{-T} F(z^0)} \rightarrow \text{itération Newton}$$
- Itérations : $z^0, z^1, \dots, z^k, \dots$ jusqu'à obtenir $F(z^*) = 0$ à une précision donnée
- **Convergence rapide, mais pas garantie** \rightarrow Technique de globalisation nécessaire

2 Algorithme SQP

- Principe
- Problème quadratique
- Résolution
- Organigramme
- Quasi-Newton
- Modification du hessien
- Globalisation

2 Algorithme SQP

Principe

- On cherche à résoudre les conditions KKT par la méthode de Newton.

- Système à résoudre : $F(z) = 0$ avec $z = \begin{pmatrix} x \\ \lambda \end{pmatrix}$

$$F(z) = \nabla L(x, \lambda) = \begin{pmatrix} \nabla f(x) + \lambda^T \nabla c(x) \\ c(x) \end{pmatrix}$$

Itération Newton

- Point courant : $z_k = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix}$

- Déplacement : $\delta z = \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix}$ vérifiant $\nabla F(z_k)^T \delta z = -F(z_k)$

$$\Leftrightarrow \nabla^2 L(x_k, \lambda_k) \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = -\nabla L(x_k, \lambda_k)$$

$$\Leftrightarrow \begin{cases} \nabla_{xx}^2 L(x_k, \lambda_k) d_x + \nabla c(x_k) d_\lambda = -\nabla_x L(x_k, \lambda_k) \\ \nabla c(x_k)^T d_x = -c(x_k) \end{cases}$$

Projet Optimisation

2 Algorithme SQP

Problème quadratique équivalent

On se place au point $(\mathbf{x}_k, \lambda_k)$.

- L'**itération de Newton** pour résoudre les conditions KKT donne le système en variables $(\mathbf{d}_x, \mathbf{d}_\lambda)$

$$\begin{cases} \nabla_{xx}^2 L(\mathbf{x}_k, \lambda_k) \mathbf{d}_x + \nabla c(\mathbf{x}_k) \mathbf{d}_\lambda = -\nabla_x L(\mathbf{x}_k, \lambda_k) \\ \nabla c(\mathbf{x}_k)^T \mathbf{d}_x = -c(\mathbf{x}_k) \end{cases}$$

- Le **problème quadratique-linéaire** (QP) en variables $\mathbf{d}_{QP} \in \mathbb{R}^n$

$$\min_{\mathbf{d}_{QP} \in \mathbb{R}^n} \frac{1}{2} \mathbf{d}_{QP}^T \nabla_{xx}^2 L(\mathbf{x}_k, \lambda_k) \mathbf{d}_{QP} + \nabla_x L(\mathbf{x}_k, \lambda_k)^T \mathbf{d}_{QP} \quad \text{sous} \quad \nabla c(\mathbf{x}_k)^T \mathbf{d}_{QP} + c(\mathbf{x}_k) = 0$$

a pour conditions d'ordre 1 le système en variables $(\mathbf{d}_{QP}, \lambda_{QP})$

$$\begin{cases} \nabla_{xx}^2 L(\mathbf{x}_k, \lambda_k) \mathbf{d}_{QP} + \nabla_x L(\mathbf{x}_k, \lambda_k) + \nabla c(\mathbf{x}_k) \lambda_{QP} = 0 \\ \nabla c(\mathbf{x}_k)^T \mathbf{d}_{QP} + c(\mathbf{x}_k) = 0 \end{cases}$$

→ multiplicateurs λ_{QP}

- **Les 2 systèmes linéaires sont identiques** en posant : $\begin{cases} \mathbf{d}_{QP} = \mathbf{d}_x \in \mathbb{R}^n \\ \lambda_{QP} = \mathbf{d}_\lambda \in \mathbb{R}^m \end{cases}$

2 Algorithme SQP

Interprétation

- L'itération de Newton au point (x_k, λ_k) équivaut à la résolution du problème quadratique :

$$\min_{d_{QP} \in \mathbb{R}^n} \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} + \nabla_x L(x_k, \lambda_k)^T d_{QP} \quad \rightarrow L \text{ à l'ordre 2 en } x$$
$$\text{sous } \nabla c(x_k)^T d_{QP} + c(x_k) = 0 \quad \rightarrow c \text{ à l'ordre 1 en } x$$

→ Minimiser un modèle quadratique local du lagrangien
sous un modèle linéaire local des contraintes

- Les itérations de Newton pour résoudre les conditions KKT sont équivalentes à la résolution d'une suite de problèmes quadratiques.

→ **Programmation Quadratique Séquentielle** (SQP)

2 Algorithme SQP

Résolution du problème quadratique

- En pratique, on remplace le problème quadratique

$$\min_{\mathbf{d}_{QP} \in \mathbb{R}^n} \frac{1}{2} \mathbf{d}_{QP}^T \nabla_{xx}^2 L(\mathbf{x}_k, \lambda_k) \mathbf{d}_{QP} + \nabla_x L(\mathbf{x}_k, \lambda_k)^T \mathbf{d}_{QP} \quad \text{sous} \quad \nabla c(\mathbf{x}_k)^T \mathbf{d}_{QP} + c(\mathbf{x}_k) = 0$$

$$\text{de solution } (\mathbf{d}_{QP}, \lambda_{QP}) \quad \rightarrow \quad \begin{cases} \mathbf{d}_x = \mathbf{d}_{QP} \\ \mathbf{d}_\lambda = \lambda_{QP} \end{cases} \quad \rightarrow \quad \begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_{QP} \\ \lambda_{k+1} = \lambda_k + \lambda_{QP} \end{cases}$$

- par le problème quadratique

$$\min_{\mathbf{d}_{QP} \in \mathbb{R}^n} \frac{1}{2} \mathbf{d}_{QP}^T \nabla_{xx}^2 L(\mathbf{x}_k, \lambda_k) \mathbf{d}_{QP} + \nabla f(\mathbf{x}_k)^T \mathbf{d}_{QP} \quad \text{sous} \quad \nabla c(\mathbf{x}_k)^T \mathbf{d}_{QP} + c(\mathbf{x}_k) = 0$$

$$\text{de solution } (\mathbf{d}_{QP}, \lambda_{QP}) \quad \rightarrow \quad \begin{cases} \mathbf{d}_x = \mathbf{d}_{QP} \\ \mathbf{d}_\lambda = \lambda_{QP} - \lambda_k \end{cases} \quad \rightarrow \quad \begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_{QP} \\ \lambda_{k+1} = \lambda_{QP} \end{cases}$$

2 Algorithme SQP

Résolution du problème quadratique

- On peut calculer explicitement la solution du problème quadratique local

$$\min_{\mathbf{d}_{QP} \in \mathbb{R}^n} \frac{1}{2} \mathbf{d}_{QP}^T \nabla_{xx}^2 L(\mathbf{x}_k, \lambda_k) \mathbf{d}_{QP} + \nabla f(\mathbf{x}_k)^T \mathbf{d}_{QP} \quad \text{sous} \quad \nabla c(\mathbf{x}_k)^T \mathbf{d}_{QP} + c(\mathbf{x}_k) = 0$$

$$\Leftrightarrow \min_{\mathbf{d}_{QP} \in \mathbb{R}^n} \frac{1}{2} \mathbf{d}_{QP}^T \mathbf{Q} \mathbf{d}_{QP} + \mathbf{g}^T \mathbf{d}_{QP} \quad \text{sous} \quad \mathbf{A} \mathbf{d}_{QP} = \mathbf{b} \quad \text{avec} \quad \begin{cases} \mathbf{Q} = \nabla_{xx}^2 L(\mathbf{x}_k, \lambda_k) \\ \mathbf{g} = \nabla f(\mathbf{x}_k) \end{cases} \quad \text{et} \quad \begin{cases} \mathbf{A} = \nabla c(\mathbf{x}_k)^T \\ \mathbf{b} = -c(\mathbf{x}_k) \end{cases}$$

- Si \mathbf{Q} est définie positive, la solution est :

$$\boxed{\begin{cases} \lambda_{QP} = (\mathbf{A} \mathbf{Q}^{-1} \mathbf{A}^T)^{-1} (\mathbf{A} \mathbf{Q}^{-1} \mathbf{g} + \mathbf{b}) \\ \mathbf{d}_{QP} = \mathbf{Q}^{-1} (\mathbf{A}^T \lambda_{QP} - \mathbf{g}) \end{cases}}$$

(Pour des problèmes de grande taille, une résolution itérative est plus efficace)

- Le déplacement à réaliser est :

$$\begin{cases} \mathbf{d}_x = \mathbf{d}_{QP} \\ \mathbf{d}_\lambda = \lambda_{QP} - \lambda_k \end{cases} \rightarrow \begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_{QP} \\ \lambda_{k+1} = \lambda_{QP} \end{cases}$$

Projet Optimisation

2 Algorithme SQP

Difficultés pratiques

L'algorithme SQP présente les mêmes défauts que la méthode de Newton.

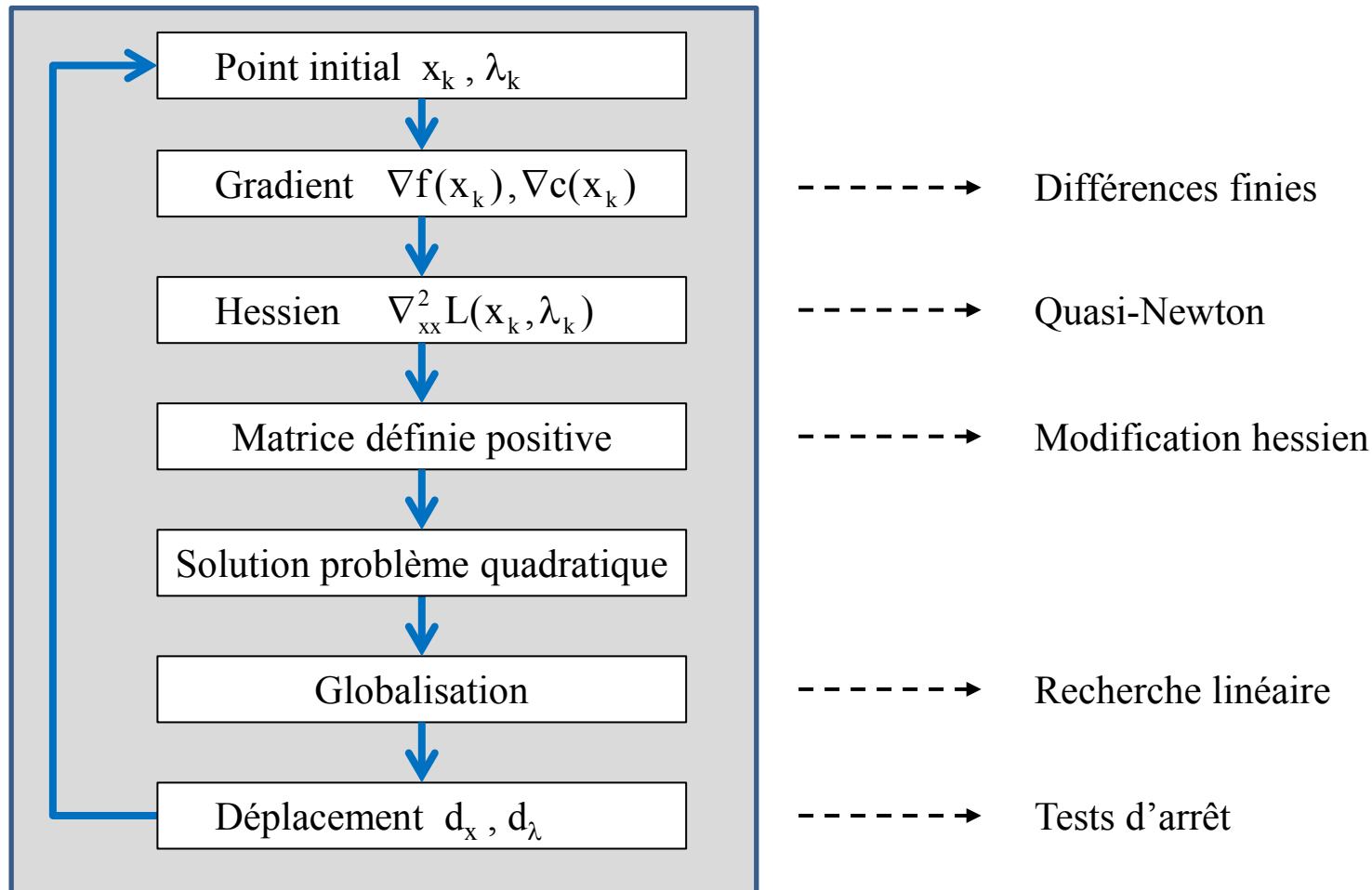
- La matrice $Q = \nabla_{xx}^2 L(x_k, \lambda_k)$ est coûteuse à calculer (hessien)
 - On remplace Q par une approximation H du hessien
 - **Méthode de quasi-Newton**
- La matrice H n'est pas forcément définie positive
 - On remplace H par une matrice définie positive « proche » H'
 - **Modification du hessien**
- Le problème quadratique n'est qu'une approximation locale du « vrai » problème
 - Le déplacement calculé ne produit pas forcément une amélioration
 - **Méthode de globalisation**

$$\begin{array}{ccc} \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} & \longrightarrow & \begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} \\ \nabla L(x_k, \lambda_k) & & \nabla L(x_{k+1}, \lambda_{k+1}) \end{array} \quad \xrightarrow{\quad ? \quad} \quad \|\nabla L(x_{k+1}, \lambda_{k+1})\| < \|\nabla L(x_k, \lambda_k)\|$$

Projet Optimisation

2 Algorithme SQP

Organigramme



2 Algorithme SQP

Quasi-Newton

- Le problème quadratique à l'itération k s'écrit :

$$\min_{\mathbf{d}_{QP} \in \mathbb{R}^n} \frac{1}{2} \mathbf{d}_{QP}^T \mathbf{Q} \mathbf{d}_{QP} + \mathbf{g}^T \mathbf{d}_{QP} \quad \text{sous} \quad \mathbf{A} \mathbf{d}_{QP} = \mathbf{b} \quad \text{avec} \quad \begin{cases} \mathbf{Q} = \nabla_{xx}^2 L(\mathbf{x}_k, \lambda_k) \\ \mathbf{g} = \nabla f(\mathbf{x}_k) \end{cases} \quad \text{et} \quad \begin{cases} \mathbf{A} = \nabla \mathbf{c}(\mathbf{x}_k)^T \\ \mathbf{b} = -\mathbf{c}(\mathbf{x}_k) \end{cases}$$

- Le hessien du lagrangien $\mathbf{Q} = \nabla_{xx}^2 L(\mathbf{x}_k, \lambda_k)$ est très coûteux à calculer.
On le remplace par une **approximation \mathbf{H}_k** construite à partir du dernier déplacement

$$\begin{cases} \mathbf{d}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1} & \rightarrow \text{variation de } \mathbf{x} \\ \mathbf{y}_{k-1} = \nabla_{\mathbf{x}} L(\mathbf{x}_k, \lambda_k) - \nabla_{\mathbf{x}} L(\mathbf{x}_{k-1}, \lambda_k) & \rightarrow \text{variation de } \nabla_{\mathbf{x}} L \text{ avec } \lambda = \lambda_k \text{ (nouvelle valeur)} \end{cases}$$

- Formule BFGS** :
$$\mathbf{H}_k = \mathbf{H}_{k-1} + \frac{\mathbf{y}_{k-1} \mathbf{y}_{k-1}^T}{\mathbf{y}_{k-1}^T \mathbf{d}_{k-1}} - \frac{\mathbf{H}_{k-1} \mathbf{d}_{k-1} \mathbf{d}_{k-1}^T \mathbf{H}_{k-1}}{\mathbf{d}_{k-1}^T \mathbf{H}_{k-1} \mathbf{d}_{k-1}} \quad \begin{array}{l} \text{si } \mathbf{y}_{k-1}^T \mathbf{d}_{k-1} > 0 \\ \text{sinon } \mathbf{H}_k = \mathbf{H}_{k-1} \end{array}$$

- Formule SR1** :
$$\mathbf{H}_k = \mathbf{H}_{k-1} + \frac{(\mathbf{y}_{k-1} - \mathbf{H}_{k-1} \mathbf{d}_{k-1})(\mathbf{y}_{k-1} - \mathbf{H}_{k-1} \mathbf{d}_{k-1})^T}{\mathbf{d}_{k-1}^T (\mathbf{y}_{k-1} - \mathbf{H}_{k-1} \mathbf{d}_{k-1})}$$

- Initialisation : $\mathbf{H}_0 = \mathbf{I}$

2 Algorithme SQP

Modification du hessien

- Le hessien du lagrangien Q (ou son approximation H) n'est pas forcément défini positif.

$$Q = \nabla_{xx}^2 L(x_k, \lambda_k) \quad \rightarrow \text{quelconque loin du minimum } (x^*, \lambda^*) \\ \rightarrow \text{peut être indéfini au minimum}$$

Pour un problème avec contrainte, seul le hessien réduit est nécessairement défini positif.

- Pour que le problème quadratique ait systématiquement une solution, on modifie le hessien Q (ou son approximation H) pour obtenir une matrice définie positive. La modification doit être « minimale » par rapport à la matrice initiale.
- 2 méthodes de modification
 - \rightarrow **factorisation de Cholesky modifiée**
 - ou \rightarrow **$H' = H + \tau I$** avec $\tau > 0$ suffisamment grand (\rightarrow valeurs propres positives)
- Lien avec la méthode de quasi-Newton :
 - BFGS \rightarrow H définie positive $\rightarrow H \neq Q$, pas de modification nécessaire
 - SR1 \rightarrow H indéfinie $\rightarrow H \approx Q$, modification nécessaire

2 Algorithme SQP

Globalisation

- L'itération SQP équivaut à une itération de la méthode de Newton, qui n'est pas robuste. Il faut contrôler la solution, et si nécessaire la modifier.

$$\begin{pmatrix} \mathbf{x}_k \\ \lambda_k \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{x}_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k \\ \lambda_k \end{pmatrix} + \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_\lambda \end{pmatrix} \quad \rightarrow \text{acceptable ?}$$

- L'objectif est de résoudre les conditions KKT : $\nabla L(\mathbf{x}^*, \lambda^*) = 0$

Il faudrait donc vérifier : $\|\nabla L(\mathbf{x}_{k+1}, \lambda_{k+1})\| < \|\nabla L(\mathbf{x}_k, \lambda_k)\|$

- L'évaluation de ∇L étant très coûteuse, on utilise à la place une **fonction mérite $F(\mathbf{x})$** .

La fonction mérite doit mesurer l'amélioration simultanée :

- du critère $f(\mathbf{x})$ à minimiser $\rightarrow \min_{\mathbf{x}} f(\mathbf{x})$
- des contraintes $\mathbf{c}(\mathbf{x})$ à annuler $\rightarrow \min_{\mathbf{x}} \|\mathbf{c}(\mathbf{x})\|_1 = \sum_{i=1}^m |c_i(\mathbf{x})|$

2 Algorithme SQP

Fonction mérite

- On peut prendre comme fonction mérite $F(x)$:

$$F(x) = f(x) + \rho \|c(x)\|_1$$

→ critère augmenté avec une **pénalisation ρ**

$$F(x) = f(x) + \lambda^T c(x) + \rho \|c(x)\|_1$$
$$= L(x, \lambda) + \rho \|c(x)\|_1$$

→ lagrangien augmenté avec une pénalisation ρ
pour λ fixé ($\lambda = \lambda_{k+1}$)

- La pénalisation ρ doit être adaptée pour que les contraintes soient respectées avec la précision voulue à la fin des itérations.
Si les contraintes sont insuffisamment respectées, la pénalisation doit être augmentée.
On peut également prévoir des pénalisations différentes sur chaque contrainte.
- La solution QP n'est acceptée que si elle fait décroître la fonction mérite.
→ $F(x_{k+1}) < F(x_k)$

Si la solution QP n'est pas acceptable, on l'utilise comme **direction de recherche** pour construire un déplacement acceptable.

2 Algorithme SQP

Recherche linéaire

- On cherche à partir de x_k un pas s dans la direction d vérifiant la **condition d'Armijo** :

$$\boxed{F(x_k + sd) < F(x_k) + c_1 s F_d'(x_k)} \quad \rightarrow \text{condition de décroissance suffisante } (c_1 = 0,1)$$

- La **direction** de recherche linéaire d est donnée par la solution du problème QP.

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} d^T Q d + g^T d \quad \text{sous } Ad = b \quad \text{avec} \quad \begin{cases} Q = \nabla_{xx}^2 L(x_k, \lambda_k) \\ g = \nabla f(x_k) \end{cases} \quad \text{et} \quad \begin{cases} A = \nabla c(x_k)^T \\ b = -c(x_k) \end{cases}$$

- La **dérivée directionnelle** $F_d'(x_k)$ de la fonction mérite dans la direction d est :

$$F_d'(x_k) = \left(\frac{dF(x_k + sd)}{ds} \right)_{s=0} = \nabla f(x_k)^T d - \rho \|c(x_k)\|_1$$

- La direction d doit être une **direction de descente** pour la fonction mérite : $F_d'(x_k) < 0$

→ vérifié si $Q > 0$ (modification du hessien)
et ρ assez grand (supérieur aux multiplicateurs λ)

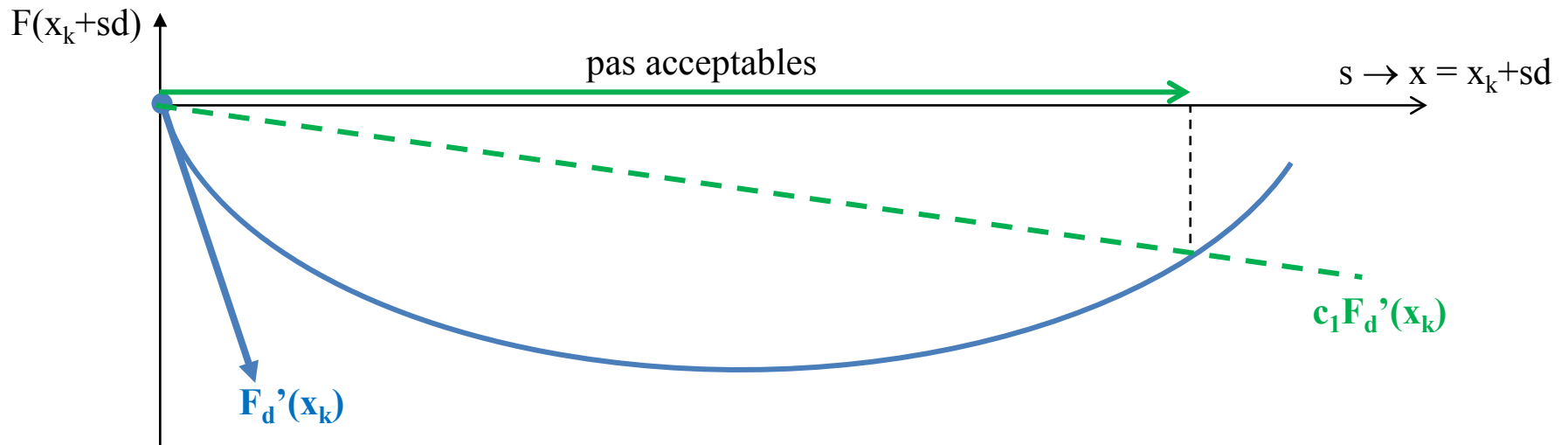
2 Algorithme SQP

Condition d'Armijo

- On cherche à partir de x_k un pas s dans la direction d vérifiant la condition d'Armijo :

$$\boxed{F(x_k + sd) < F(x_k) + c_1 s F_d'(x_k)}$$

→ condition de décroissance suffisante ($c_1 = 0,1$)



- Réglage du pas :
 - $s_0 = 1$ → pas de Newton (solution QP)
 - $s_{j+1} = s_j/2$ → jusqu'à vérifier la condition d'Armijo