

---

# Optimisation paramétrique

---

**Version 1.0**

23 novembre 2010

Michel Llibre

---

Ref. DCSD-2009\_008-NOT-009-1.0





# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Conventions sur les notations</b>	<b>7</b>
2.1	Expression matricielle des gradients . . . . .	7
2.1.1	Gradient d'une forme linéaire . . . . .	7
2.1.2	Gradient d'une forme quadratique . . . . .	8
2.2	Expression des matrices jacobiniennes . . . . .	8
<b>3</b>	<b>Conditions d'optimalité</b>	<b>11</b>
3.1	Minimum d'un fonction scalaire sans contraintes . . . . .	11
3.2	Minimum avec contraintes égalités - Lagrangien - Paramètres de Lagrange . . . . .	12
3.3	Minimum avec contraintes inégalités - Paramètres de Kuhn et Tucker . . . . .	12
3.4	Formule du minimax . . . . .	13
<b>4</b>	<b>Programmation linéaire</b>	<b>15</b>
4.1	La forme standard . . . . .	15
4.2	Méthode du simplexe . . . . .	16
4.2.1	Obtention d'une solution de base réalisable initiale . . . . .	16
4.2.2	Passage d'un sommet à l'autre . . . . .	17
4.2.3	Règle de sélection de la variable entrante . . . . .	18
4.2.4	Dégénérescence . . . . .	19
4.2.5	Règle d'anti-cyclage . . . . .	19
4.3	Passage à la forme standard . . . . .	19
4.3.1	Variables négatives . . . . .	19
4.3.2	Variables non contraintes . . . . .	19
4.3.3	Contraintes inégalités . . . . .	19
4.4	La dualité . . . . .	20
<b>5</b>	<b>Optimisation positive</b>	<b>23</b>
5.1	Enoncé du problème . . . . .	23
5.2	Théorèmes de l'hyperplan séparateur . . . . .	23
5.3	Le théorème de Farkas . . . . .	24
5.4	Matrices opérationnelles et noyau coopératif . . . . .	24
5.4.1	Cas de la redondance simple : $n = m + 1$ . . . . .	25

5.4.2	Cas de la redondance multiple . . . . .	26
5.5	Résolution en redondance multiple . . . . .	27
5.5.1	Critère linéaire - Résolution par le simplexe . . . . .	28
5.5.2	Critère non-linéaire - Méthode de point intérieur . . . . .	28
<b>6</b>	<b>Programmation quadratique</b>	<b>29</b>
6.1	Optimum sans contraintes inégalités : . . . . .	29
6.2	Optimum avec contraintes inégalités . . . . .	30
6.2.1	Saturation d'une seule contrainte . . . . .	30
6.2.2	Saturation de plusieurs contraintes . . . . .	33
6.2.3	Stratégie de saturation . . . . .	34
<b>7</b>	<b>PNL sans contraintes</b>	<b>37</b>
7.1	Cas monovariante . . . . .	37
7.2	Cas multivariante . . . . .	37
7.2.1	Méthodes d'ordre zéro . . . . .	37
7.2.2	Méthodes d'ordre 1 : Algorithmes de descente . . . . .	38
7.2.3	Méthode d'ordre 2 . . . . .	39
7.2.4	Méthodes géométriques . . . . .	40
<b>8</b>	<b>PNL avec contraintes (A Faire)</b>	<b>41</b>
8.1	Méthode du gradient projeté . . . . .	41
8.2	Fonctions de pénalisation . . . . .	42

# Chapitre 1

## Introduction

Nous présentons dans ce document des techniques utilisées pour résoudre des problèmes d'optimisation paramétrique. Concrètement, il s'agit de trouver un ensemble réduit de paramètres optimaux qui sont soumis à un certain nombre de contraintes. Nous utilisons le qualificatif réduit pour distinguer les problèmes étudiés des problèmes de commandes ou trajectoires optimales qui recherchent un ensemble infini (ou très grand) de paramètres qui sont les composantes de tous les points des trajectoires recherchées, bien que par discrétisation du temps, on peut ramener la résolution des problèmes de trajectoires optimales à de l'optimisation paramétrique en très grande dimension.

Généralement, on arrive à résoudre les problèmes d'optimisation paramétrique lorsqu'ils sont convexes, c'est-à-dire lorsque :

- **l'ensemble**  $\mathcal{C}$  des valeurs admissibles des paramètres inconnus  $\mathbf{x}$  **est convexe** : L'ensemble  $\mathcal{C} \subset \mathbb{R}^n$  est dit convexe, si pour tout couple de points  $\mathbf{x}_1$  et  $\mathbf{x}_2 \in \mathcal{C}$ , le segment  $[\mathbf{x}_1, \mathbf{x}_2]$  est entièrement contenu dans  $\mathcal{C}$ ,
- **la fonction** critère à minimiser  $f(\mathbf{x}) \in \mathbb{R}$  **est convexe** pour  $\mathbf{x} \in \mathcal{C}$  convexe, ce qui est le cas si :

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C} \text{ et } \forall \lambda \in [0, 1] \text{ on a : } f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2)$$

autrement dit si  $f$  est toujours inférieure à son approximation linéaire.

On appelle **programmation mathématique** le problème qui consiste à trouver  $\mathbf{x} \in \mathcal{C} \subset \mathbb{R}^n$  qui minimise la fonction  $f(\mathbf{x}) \in \mathbb{R}$ . On l'écrit :

$$\min f(\mathbf{x}) \in \mathbb{R}, \quad \mathbf{x} \in \mathcal{C} \subset \mathbb{R}^n$$

- $f(\mathbf{x})$  est appelé la *fonction objectif*
- Le sous-ensemble  $\mathcal{C}$  de  $\mathbb{R}^n$  est généralement spécifié par un ensemble de contraintes égalités et/ou inégalités.
- Tout  $\mathbf{x} \in \mathcal{C}$  est appelé *solution admissible*.

L'intérêt de la convexité repose sur le fait que lorsque  $f(\mathbf{x})$  est convexe pour tout  $\mathbf{x} \in \mathcal{C}$ , tout minimum local est minimum global. Dans le cas contraire, on ne sait pas conclure quant à l'optimalité globale des solutions qui remplissent les conditions d'optimalité du premier ordre.

Parmi les problèmes de programmation mathématique, nous aborderons :

- la programmation linéaire, développée par Georges Dantzig en 1940, qui considère le cas où  $f(\mathbf{x})$  est linéaire et où les contraintes sont linéaires,
- la programmation quadratique qui considère le cas où  $f(\mathbf{x})$  présente des termes quadratiques, toujours avec des contraintes linéaires,
- la programmation non linéaire qui traite les cas où  $f(\mathbf{x})$  et/ou les contraintes présentent des parties non-linéaires.

Nous n'aborderons la programmation stochastique qui traite les cas où certaines contraintes dépendent de variables aléatoires.

Le premier chapitre présente les conventions sur les notations : comment sont notés les scalaires, les vecteurs, ... et comment sont notées les dérivations de ces éléments.

Le deuxième chapitre présente les conditions d'optimalité et en particulier les notions de lagrangien, de paramètres de Lagrange, et de paramètres de Kuhn et Tucker ou de Karush-Kuhn-Tucker.

Le troisième chapitre présente la programmation linéaire, c'est-à-dire la méthode utilisée pour optimiser un critère linéaire en fonction des paramètres inconnus, avec des contraintes linéaires sur ces paramètres.

Le quatrième chapitre présente quelques résultats théoriques sur l'optimisation positive. Nous appelons ainsi un problème pour lequel les paramètres inconnus doivent tous être de même signe (positifs par exemple). Ce problème se présente par exemple lorsqu'on recherche avec quelles tensions il faut tirer sur  $n \geq 7$  câbles pour imposer un torseur d'effort donné (6 composantes, 3 en force et 3 en moment) à un solide attaché par ces  $n$  câbles, par exemple pour compenser son poids et le tenir en équilibre, ou pour lui communiquer une accélération désirée. Tous les câbles doivent être tendus (tensions positives) pour que la suspension de l'objet opérationnelle (contraintes inégalités), la combinaison des forces de tension devant être égale au torseur résultant désiré (contraintes égalités). On se pose le problème de l'existence de solutions et du choix de la meilleure, lorsqu'il en existe.

Le cinquième chapitre présente un exemple de résolution par la programmation quadratique.

Le dernier chapitre présente quelques techniques utilisées en programmation non-linéaire, sans contraintes.

## Chapitre 2

# Conventions sur les notations

Les scalaires sont en général noté  $s$  avec une lettre minuscule italique. Les vecteurs (matrices lignes ou colonnes) sont généralement notés avec une lettre grasse, généralement minuscule :  $\mathbf{l} = \mathbf{v}^T$ . Les matrices sont généralement notées avec une lettre majuscule grasse. L'expression  $s = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}$  réunit ces 3 notations.

### 2.1 Expression matricielle des gradients

Considérons une fonction scalaire  $s$  de plusieurs variables :

$$s = S(x_1, x_2, \dots, y_1, y_2, \dots, \dots) = S(\mathbf{x}, \mathbf{y}, \dots)$$

Le gradient de  $s$  par rapport au vecteur  $\mathbf{x}$  de dimension  $n$  sera noté  $s_{\mathbf{x}}$ ,  $S_{\mathbf{x}}$  ou  $\nabla_{\mathbf{x}} s$  :

$$\nabla_{\mathbf{x}} s = \frac{\partial S}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial S}{\partial x_1} \\ \frac{\partial S}{\partial x_2} \\ \vdots \\ \frac{\partial S}{\partial x_n} \end{pmatrix}$$

Si  $s = s_0$  est une constante par rapport à  $\mathbf{x}$ , alors  $\nabla_{\mathbf{x}} s = \mathbf{0}$ .

**Remarque** : Nous traitons le gradient  $\nabla_{\mathbf{x}} s$  comme un *vecteur colonne*, contrairement à un usage fréquent que l'on trouve dans les ouvrages purement mathématiques où il est traité comme un vecteur ligne.

#### 2.1.1 Gradient d'une forme linéaire

Considérons :

$$s = s_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n = s_0 + \mathbf{b}^T \mathbf{x} = s_0 + \mathbf{x}^T \mathbf{b}$$

Alors :

$$\nabla_{\mathbf{x}} s = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \mathbf{b}$$

soit :

$$\nabla_{\mathbf{x}} (s_0 + \mathbf{x}^T \mathbf{b}) = \mathbf{b}$$

### 2.1.2 Gradient d'une forme quadratique

Considérons :

$$s = s_0 + \mathbf{x}^T \mathbf{b} + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_i x_j = s_0 + \mathbf{x}^T \mathbf{b} + \frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x}$$

On peut toujours considérer que  $\mathbf{B}$  est symétrique. En effet :

$$s = \frac{s + s^T}{2} = s_0 + \mathbf{x}^T \mathbf{b} + \frac{1}{2} \mathbf{x}^T \left( \frac{\mathbf{C} + \mathbf{C}^T}{2} \right) \mathbf{x}$$

$$s = s_0 + \mathbf{x}^T \mathbf{b} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} \quad \text{avec } \mathbf{A} \text{ symétrique : } \mathbf{A} = \frac{\mathbf{C} + \mathbf{C}^T}{2}$$

Il vient :

$$\begin{aligned} \frac{\partial S}{\partial x_k} &= b_k + \frac{1}{2} \sum_{j=1}^n c_{kj} x_j + \frac{1}{2} \sum_{i=1}^n c_{ik} x_i = b_k + \sum_{i=1}^n \frac{c_{ik} + c_{kj}}{2} x_i \\ \frac{\partial S}{\partial x_k} &= b_k + \sum_{i=1}^n a_{ik} x_i \end{aligned}$$

D'où :

$$\nabla_x (s_0 + \mathbf{x}^T \mathbf{b} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}) = \mathbf{b} + \mathbf{A} \mathbf{x}$$

En résumé, dans le cas d'une forme linéaire  $s = \mathbf{x}^T \mathbf{b} = \mathbf{b}^T \mathbf{x}$ , on a  $\nabla_x s = \mathbf{b}$  et dans le cas d'une forme quadratique  $s = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}$  on a  $\nabla_x s = \mathbf{A} \mathbf{x}$ . Dans le cas d'une forme linéaire le vecteur gradient est le facteur de  $\mathbf{x}^T$  et dans le cas d'une forme quadratique c'est le facteur de  $\frac{1}{2} \mathbf{x}^T$ .

## 2.2 Expression des matrices jacobiennes

Soit une fonction vectorielle différentiable  $\mathbf{f}(\mathbf{x})$  de dimension  $m$  sur un espace de point  $\mathbf{x} \in \mathbb{R}^n$  de dimension  $n$ . Notons  $x_1, x_2, \dots, x_n$ , les coordonnées de  $\mathbf{x}$  et  $f_1, f_2, \dots, f_m$  les composantes de  $\mathbf{f}(\mathbf{x})$ .

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_m(x_1, x_2, \dots, x_n) \end{pmatrix}$$

Par convention, nous noterons :

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$$

la matrice formée des éléments suivants :

$$\mathbf{J} = \begin{pmatrix} (\nabla_x f_1)^T \\ (\nabla_x f_2)^T \\ \vdots \\ (\nabla_x f_m)^T \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Lorsque  $m = n$ , on appelle jacobien le scalaire  $j = \det(\mathbf{J})$ .



**Remarque :** Si  $m = 1$ ,  $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$  est une matrice ligne à  $n$  éléments, alors qu'avec nos conventions matricielles, l'opérateur  $\frac{\partial}{\partial \mathbf{x}}$  transforme un scalaire en un vecteur colonne. **C'est une incohérence de nos notations** (que l'on trouve dans de nombreux autres ouvrages). Pour se garder des erreurs que risque d'entraîner cette incohérence, il faut se souvenir que  $\mathbf{f}$  est de nature vectorielle et non scalaire, même quand elle n'a qu'une composante. Et qu'en conséquence  $\mathbf{J}$  ne représente pas le gradient d'un scalaire (que l'on écrit en colonne), mais le gradient d'une composante (que l'on écrit en ligne).



## Chapitre 3

# Conditions d'optimalité

Nous introduisons dans cette section le Lagrangien et les paramètres de Kuhn et Tucker qui sont des outils essentiels pour l'optimisation.

### 3.1 Minimum d'une fonction scalaire sans contraintes

Considérons une fonction scalaire  $r(\mathbf{x})$  du point  $\mathbf{x}$  de  $\mathbb{R}^n$ , différentiable par rapport à  $\mathbf{x}$ . Soit  $\hat{\mathbf{x}}$  un minimum relatif de  $r(\mathbf{x})$ . Pour alléger les notations nous noterons :

- $r_{\mathbf{x}}$  le gradient  $\nabla_{\mathbf{x}} r$
- $r_{\mathbf{xx}}$  la matrice carrée symétrique<sup>1</sup> de composantes  $\frac{\partial^2 r}{\partial x_i \partial x_j}$  également notée  $\nabla_{\mathbf{x}}^2 r$ .

Quelle que soit la direction du vecteur  $\delta \mathbf{x}$  on a :

$$\left. \begin{array}{l} r(\hat{\mathbf{x}}) \leq r(\hat{\mathbf{x}} + \delta \mathbf{x}) \rightarrow r_{\hat{\mathbf{x}}}^T \delta \mathbf{x} \geq 0 \\ r(\hat{\mathbf{x}}) \leq r(\hat{\mathbf{x}} - \delta \mathbf{x}) \rightarrow r_{\hat{\mathbf{x}}}^T \delta \mathbf{x} \leq 0 \end{array} \right\} \rightarrow r_{\hat{\mathbf{x}}}^T \delta \mathbf{x} = 0 \rightarrow r_{\hat{\mathbf{x}}} = \mathbf{0}$$

**Conditions nécessaires d'optimalité du premier ordre :**

Pour que  $\hat{\mathbf{x}}$  soit un minimum relatif de  $r(\mathbf{x})$  il est nécessaire que  $r_{\hat{\mathbf{x}}} = \mathbf{0}$ .

De plus si  $r(\mathbf{x})$  est doublement différentiable, on a :

$$r(\hat{\mathbf{x}}) \leq r(\hat{\mathbf{x}} + \delta \mathbf{x}) = r(\hat{\mathbf{x}}) + r_{\hat{\mathbf{x}}}^T \delta \mathbf{x} + \frac{1}{2} \delta \mathbf{x}^T r_{\hat{\mathbf{x}}\hat{\mathbf{x}}} \delta \mathbf{x}$$

D'où, compte tenu du fait que  $r_{\hat{\mathbf{x}}} = \mathbf{0}$  :

$$\delta \mathbf{x}^T r_{\hat{\mathbf{x}}\hat{\mathbf{x}}} \delta \mathbf{x} \geq 0 \Leftrightarrow r_{\hat{\mathbf{x}}\hat{\mathbf{x}}} \text{ définie non négative}$$

**Conditions du deuxième ordre :**

Pour que  $\hat{\mathbf{x}}$  soit un minimum relatif de  $r(\mathbf{x})$  il est nécessaire que  $r_{\hat{\mathbf{x}}} = \mathbf{0}$  et que  $r_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$  soit définie non négative.

Si  $r_{\hat{\mathbf{x}}} = \mathbf{0}$  et si  $r_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$  est définie positive,  $\hat{\mathbf{x}}$  est un minimum relatif strict de  $r(\mathbf{x})$ .

1. Cette matrice porte le nom de hessien.

### 3.2 Minimum avec contraintes égalités - Lagrangien - Paramètres de Lagrange

Supposons que  $\mathbf{x}$  soit contraint par les  $p$  égalités suivantes :

$$\mathbf{g}(\mathbf{x}) = 0$$

Si  $\hat{\mathbf{x}}$  est un minimum relatif de  $r(\mathbf{x})$  vérifiant les  $p$  équations  $\mathbf{g}(\hat{\mathbf{x}}) = \mathbf{0}$  et  $r_{\hat{\mathbf{x}}}^T \delta \mathbf{x} = 0$  quel que soit  $\delta \mathbf{x}$  tel que  $\mathbf{g}(\hat{\mathbf{x}} + \delta \mathbf{x}) = \mathbf{0}$ . Les  $\delta \mathbf{x}$  autorisés sont donc tels que  $\mathbf{g}_{\hat{\mathbf{x}}} \delta \mathbf{x} = 0$  ( $\mathbf{g}_{\hat{\mathbf{x}}}$  est la matrice jacobienne de  $\mathbf{g}$  au point  $\hat{\mathbf{x}}$ ). Ils engendrent un sous-espace tangent  $\mathcal{N}$  de dimension  $n - p$  (noyau de  $\mathbf{g}_{\hat{\mathbf{x}}}$ ). Le vecteur  $r_{\hat{\mathbf{x}}}$  doit être orthogonal à  $\mathcal{N}$ .  $r_{\hat{\mathbf{x}}}$  est donc limité au sous espace orthogonal complémentaire de dimension  $p$ , engendré par les vecteurs lignes de  $\mathbf{g}_{\hat{\mathbf{x}}}$  qui sont orthogonaux à  $\mathcal{N}$ .  $r_{\hat{\mathbf{x}}}$  est donc combinaison linéaire des  $p$  vecteurs lignes  $\mathbf{g}_{\hat{\mathbf{x}}}$ . En notant  $-\lambda$  le vecteur des coefficients de cette combinaison linéaire, il vient :

$$r_{\hat{\mathbf{x}}} + \mathbf{g}_{\hat{\mathbf{x}}}^T \lambda = \mathbf{0}$$

Cette expression est la condition nécessaire d'optimalité du premier ordre pour la fonction relative à la variable  $\mathbf{x}$ . On l'appelle le *lagrangien* :

$$\mathcal{L}(\mathbf{x}, \lambda) = r(\mathbf{x}) + \mathbf{g}^T(\mathbf{x}) \lambda = r(\mathbf{x}) + \lambda^T \mathbf{g}(\mathbf{x})$$

La condition nécessaire du premier ordre relative à la variable  $\lambda$  redonne la contrainte  $\mathbf{g}(\mathbf{x}) = 0$ .

Les coefficients de  $\lambda$  sont appelés des *paramètres de Lagrange*.

La condition du deuxième ordre impose que la restriction de  $\mathcal{L}_{\mathbf{xx}}$  au sous-espace  $\mathcal{N}$  soit définie non négative, ce qui est équivalent à :

$$\text{Toutes les racines } s \text{ de } \begin{vmatrix} s\mathbf{I} - \mathcal{L}_{\mathbf{xx}} & \mathbf{h}_{\hat{\mathbf{x}}}^T \\ \mathbf{h}_{\hat{\mathbf{x}}} & \mathbf{0} \end{vmatrix} = 0 \text{ sont non négatives.}$$

**Remarque** : Si  $\hat{\mathbf{x}}$  est optimal, on a  $\mathbf{g}(\hat{\mathbf{x}}) = 0$  d'où  $\mathcal{L}(\hat{\mathbf{x}}, \lambda) = r(\hat{\mathbf{x}})$

### 3.3 Minimum avec contraintes inégalités - Paramètres de Kuhn et Tucker

Supposons  $\mathbf{x}$  soit contraint par les  $p$  inégalités suivantes :

$$\mathbf{h}(\mathbf{x}) \leq 0$$

En ajoutant des quantités positives inconnues  $y_i^2$  elles sont ramenées à  $p$  contraintes égalités :

$$h_i(\mathbf{x}) + y_i^2 = 0$$

Les conditions du premier ordre relatives au lagrangien :

$$\mathcal{L}(\mathbf{x}, \mu, \mathbf{y}) = r(\mathbf{x}) + \sum \mu_i (h_i(\mathbf{x}) + y_i^2)$$

s'écrivent :

$$\mathcal{L}_{\mathbf{x}} = 0 \rightarrow r_{\mathbf{x}} + \mathbf{h}_{\mathbf{x}}^T \mu = 0$$

$$\mathcal{L}_{\mathbf{y}} = 0 \rightarrow \mu_i y_i = 0$$

Elles sont équivalentes à considérer le lagrangien sans terme  $\mathbf{y}$  :

$$\mathcal{L}(\mathbf{x}, \mu) = r(\mathbf{x}) + \mu^T \mathbf{h}(\mathbf{x})$$

et envisager les  $2^p$  cas :

1. contrainte inactive ( $h_i(\mathbf{x})$  est ignoré) : résoudre avec  $\mu_i = 0$ , puis vérifier a posteriori que  $h_i(\mathbf{x}) \leq 0$ ,
2. contrainte active ou saturée : résoudre avec  $h_i(\mathbf{x}) = 0$ , puis vérifier a posteriori que  $\mu_i \geq 0$ .

Cette deuxième condition vient du fait qu'il faut vérifier qu'il n'y ait pas de  $\delta\mathbf{x}$  admissible tel que  $r(\hat{\mathbf{x}} + \delta\mathbf{x})$  soit inférieur à  $r(\hat{\mathbf{x}})$  avec  $h_i(\hat{\mathbf{x}} + \delta\mathbf{x}) < 0$ . Or  $h_i(\hat{\mathbf{x}} + \delta\mathbf{x}) = h_i(\hat{\mathbf{x}}) + h_i^T \delta\mathbf{x} = h_i^T \delta\mathbf{x} < 0$ . De plus  $r(\hat{\mathbf{x}} + \delta\mathbf{x}) = r(\hat{\mathbf{x}}) + r_x^T \delta\mathbf{x} \geq r(\hat{\mathbf{x}})$  implique  $r_x^T \delta\mathbf{x} \geq 0$ . Or  $\mathcal{L}_x = r_x + \mathbf{h}_x^T \mu = 0$  implique  $\delta\mathbf{x}^T r_x + \delta\mathbf{x}^T \mathbf{h}_x^T \mu = 0$ , implique  $\delta\mathbf{x}^T \mathbf{h}_x^T \mu \leq 0$  quel que soit  $\delta\mathbf{x}$  admissible, c'est-à-dire tel que  $\mathbf{h}_x \delta\mathbf{x} \leq 0$ . Il faut donc que tous les  $\mu_i \geq 0$ .

Les  $\mu_i$  sont appelés *paramètres de Kuhn et Tucker*<sup>2</sup>.

La condition du deuxième ordre est identique à celle utilisée avec les paramètres de Lagrange, en ne faisant intervenir que les paramètres des contraintes saturées.

**Remarque :**

1. Si  $\hat{\mathbf{x}}$  est optimal,  $\mu_i y_i = 0$  implique  $\mu_i h_i(\hat{\mathbf{x}}) = 0$ . Il en résulte que  $\mu^T \mathbf{h}(\hat{\mathbf{x}}) = 0$ , d'où  $\mathcal{L}(\hat{\mathbf{x}}, \mu) = r(\hat{\mathbf{x}})$ .
2. Les paramètres de Kuhn et Tucker associés aux contraintes saturées doivent être du *signe opposé au signe désiré pour la fonction dans le cas d'une minimisation* et du même signe dans le cas d'une maximisation en supposant qu'ils sont introduits dans le lagrangien par une *addition* du terme paramètre qui multiplie la fonction.

La remarque 2 est une règle simple qu'il faut mémoriser pour éviter d'avoir à retrouver le chemin assez complexe qui permet de l'établir.

### 3.4 Formule du minimax

Si  $\hat{\mathbf{x}}$  et  $\hat{\mu}$  sont localement tels que :

$$\mathcal{L}(\hat{\mathbf{x}}, \hat{\mu}) = \min_{\mathbf{x}} \left\{ \max_{\mu_i \geq 0} \mathcal{L}(\mathbf{x}, \mu) \right\}$$

le point  $\hat{\mathbf{x}}$  est un *minimum local* de  $r(\mathbf{x})$  compte tenu des contraintes  $\mathbf{h}(\mathbf{x}) \leq 0$ . En effet la formule du minimax implique que :

$$\mathcal{L}(\hat{\mathbf{x}}, \mu) \leq \mathcal{L}(\hat{\mathbf{x}}, \hat{\mu}) \leq \mathcal{L}(\mathbf{x}, \hat{\mu})$$

quel que soit  $\hat{\mathbf{x}}$  et  $\mu_i \geq 0$ . Alors :

1.  $h_i(\hat{\mathbf{x}}) \leq 0$ . Par l'absurde : Considérons  $\mu_i = \hat{\mu}_i$  si  $h_i(\hat{\mathbf{x}}) \leq 0$  et  $\mu_k = \hat{\mu}_k + 1$  si  $h_k(\hat{\mathbf{x}}) > 0$ . Il en résulte que :  
 $\mathcal{L}(\hat{\mathbf{x}}, \hat{\mu}) - \mathcal{L}(\hat{\mathbf{x}}, \mu) = -\sum h_k(\hat{\mathbf{x}}) < 0$  en contradiction avec l'hypothèse. D'où  $\hat{\mathbf{x}}$  respecte les contraintes.
2. Si  $\hat{\mathbf{x}}$  respecte les contraintes,  $r(\mathbf{x}) = \mathcal{L}(\mathbf{x}, \hat{\mu}) - \hat{\mu}^T \mathbf{h}(\mathbf{x})$  avec  $\hat{\mu}^T \mathbf{h}(\mathbf{x}) \leq 0$  implique  $r(\mathbf{x}) \geq \mathcal{L}(\mathbf{x}, \hat{\mu})$ .  
 Or :  
 $r(\hat{\mathbf{x}}) = \mathcal{L}(\hat{\mathbf{x}}, \mathbf{0}) \leq \mathcal{L}(\hat{\mathbf{x}}, \hat{\mu}) \leq \mathcal{L}(\mathbf{x}, \hat{\mu}) \leq r(\mathbf{x})$  ce qui démontre la formule.

Ainsi toute solution au problème du minimax est solution du problème de minimum avec contraintes inégalités (ou égalités). Réciproquement, la solution au problème de minimum avec contraintes égalités est solution du problème du minimax.

---

2. Dans les ouvrages récents et en particulier de langue anglaise, les paramètres de Kuhn et Tucker sont appelés paramètres de Karush-Kuhn-Tucker en reconnaissance du fait que William Karush fut le premier à les utiliser dans sa thèse, avant leur diffusion par la publication de Kuhn et Tucker.



## Chapitre 4

# Programmation linéaire

On appelle programme linéaire un problème d'optimisation (minimisation ou maximisation) d'un critère linéaire, en présence de contraintes égalités ou inégalités également linéaires.

### 4.1 La forme standard

Tout programme linéaire peut se mettre sous la forme standard suivante : Trouver le vecteur  $\mathbf{x}$  de  $\mathbb{R}^n$  soumis à  $\mathbf{Ax} = \mathbf{b}$  et  $\mathbf{x} \geq \mathbf{0}$ , qui minimise le produit scalaire  $\mathbf{c}^T \mathbf{x}$  :

$$\begin{aligned} \min_{\mathbf{x}} \quad & f = \mathbf{c}^T \mathbf{x} \\ \text{s.c.} \quad & \begin{cases} \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases} \end{aligned} \quad (4.1)$$

Nous supposons en outre que la matrice  $\mathbf{A}$  est de rang complet :  $\text{rang}(\mathbf{A}) = m$  nombre de lignes de  $\mathbf{A}$  et que  $\mathbf{b} \geq \mathbf{0}$  (quitte à changer de signe certaines lignes de  $\mathbf{A}$ ). La matrice  $\mathbf{A}$  possède un noyau orthonormé  $\mathbf{N}$  de dimension  $n \times r$  avec  $r = n - m$ . Une factorisation QRE (cf. paragraphe 3.7 du mémo "Algorithmes numériques pour l'algèbre linéaire") permettrait de trouver une permutation  $\mathbf{E}$  des colonnes de  $\mathbf{A}$  qui sélectionne  $m$  colonnes indépendantes de  $\mathbf{A}$ . En faisant subir à  $\mathbf{x}$ , à  $\mathbf{c}$  et à  $\mathbf{A}$  cette permutation, on obtient deux vecteurs  $\mathbf{x}_B$  et  $\mathbf{x}_N$  de dimensions respectives  $m$  et  $r$ , deux vecteurs  $\mathbf{c}_B$  et  $\mathbf{c}_N$  de dimensions respectives  $m$  et  $r$  et deux matrices  $\mathbf{A}_B$  et  $\mathbf{A}_N$  de dimensions respectives  $m \times m$  et  $m \times r$ , avec  $\det(\mathbf{A}_B) \neq 0$ , tels que :

$$\mathbf{AE} = [ \mathbf{A}_B \quad \mathbf{A}_N ] \quad ; \quad \mathbf{E}^T \mathbf{x} = \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{pmatrix} \quad ; \quad \mathbf{c}^T \mathbf{E} = ( \mathbf{c}_B^T \quad \mathbf{c}_N^T )$$

D'où :

$$\begin{aligned} f &= \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \\ \mathbf{A}_B \mathbf{x}_B + \mathbf{A}_N \mathbf{x}_N &= \mathbf{b} \\ \mathbf{x}_B &= \mathbf{A}_B^{-1} (\mathbf{b} - \mathbf{A}_N \mathbf{x}_N) \end{aligned} \quad (4.2)$$

Pour simplifier nous supposons dans ce qui suit que ces permutations sont déjà effectuées, c'est-à-dire que  $\mathbf{AE} = \mathbf{A}$ , que  $\mathbf{E}^T \mathbf{x} = \mathbf{x}$ , et que  $\mathbf{c}^T \mathbf{E} = \mathbf{c}^T$ .

Toute matrice sous-matrice  $\mathbf{A}_B$ , constituée de  $m$  colonnes de  $\mathbf{A}$ , qui est régulière est appelée *base* de  $\mathbf{A}$ .

Le vecteur :

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{pmatrix} \text{ avec } \begin{cases} \mathbf{x}_B = \mathbf{A}_B^{-1} \mathbf{b} \\ \mathbf{x}_N = \mathbf{0} \end{cases}$$

constitue une *solution de base*. Si de plus  $\mathbf{x}_B = \mathbf{A}_B^{-1} \mathbf{b} \geq \mathbf{0}$ , c'est une *solution de base réalisable*.

Les composantes de  $\mathbf{x}_B$  sont traditionnellement appelées *variables de base* et celles de  $\mathbf{x}_N$  *variables hors base*.

L'ensemble des solutions  $\mathbf{x}$  réalisables de  $\mathbb{R}^n$  constituent un polyèdre convexe, intersection des  $m$  hyperplans  $\mathbf{A}\mathbf{x} = \mathbf{b}$  et des  $n$  demi-espaces  $\mathbf{x} \geq \mathbf{0}$ . Cet ensemble peut être vide, limité à un point, à un segment, ..., un hyper-volume fermé ou illimité. En conséquence, la solution qui minimise  $\mathbf{c}^T \mathbf{x}$  est soit :

1. inexistante,
2. unique,
3. multiple, sur un segment, une surface, ...
4. non bornée.

Cette solution qui minimise  $\mathbf{c}^T \mathbf{x}$  est forcément un des sommets du polyèdre. Ces sommets sont à l'intersection de  $n$  hyperplans qui sont les  $m$  contraintes  $\mathbf{A}\mathbf{x} = \mathbf{b}$  et  $n - m$  hyperplans parmi les  $n$  hyperplans  $\mathbf{x} = \mathbf{0}$ . Ces sommets ont donc au moins  $r = n - m$  coordonnées nulles. Chaque sommet est donc caractérisé par  $m$  variables de base  $\mathbf{x}_B \geq \mathbf{0}$  et  $r$  variables hors bases  $\mathbf{x}_N$  nulles.

En multipliant le système (4.2) par  $\mathbf{A}_B^{-1}$  on le met sous une *forme simpliciale* :

$$\mathbf{I}\mathbf{x}_B + \mathbf{R}\mathbf{x}_N = \mathbf{b}' \text{ avec } \begin{cases} \mathbf{R} = \mathbf{A}_B^{-1} \mathbf{A}_N \\ \mathbf{b}' = \mathbf{A}_B^{-1} \mathbf{b} \end{cases}$$

## 4.2 Méthode du simplexe

La méthode du simplexe (ou *méthode du pivot*, ou *méthode de Dantzig*) itère d'un sommet à un autre sommet adjacent de telle manière que le critère  $\mathbf{c}^T \mathbf{x}$  diminue (ou plus exactement n'augmente pas) à chaque saut. Comme il y a un maximum de  $C_n^r$  sommets, la méthode converge en un nombre fini de pas, à condition de ne pas cycler, sur un ensemble de sommets où le revenu  $\mathbf{c}^T \mathbf{x}$  est stationnaire. Ceci peut se produire en un sommet où une ou plusieurs composantes de  $\mathbf{x}_B$  sont nulles. Pour éviter ces cycles des règles spécifiques d'*anti-cyclage* sont à appliquer.

La méthode du simplexe suppose de détenir une solution de base réalisable initiale, ce qui est le cas si  $\mathbf{b}' \geq \mathbf{0}$ .

### 4.2.1 Obtention d'une solution de base réalisable initiale

Si on ne possède pas de solution de base réalisable initiale pour  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , avec  $\mathbf{x} \geq \mathbf{0}$ , on peut en calculer une par *la méthode des deux phases*. On considère le programme linéaire préliminaire suivant :

Trouver le vecteur  $\mathbf{X} = \begin{pmatrix} \mathbf{z} \\ \mathbf{x} \end{pmatrix} \geq \mathbf{0}$ , soumis aux contraintes égalités  $\mathbf{I}\mathbf{z} + \mathbf{A}\mathbf{x} = \mathbf{b}$  qui minimise  $\sum_{i=1}^m z_i$ .

Ce programme linéaire préliminaire est sous forme simpliciale, et il a pour solution de base réalisable  $\begin{pmatrix} \mathbf{z} = \mathbf{b} \\ \mathbf{x} = \mathbf{0} \end{pmatrix}$  (ce qui explique le choix  $\mathbf{b} \geq \mathbf{0}$ ).

Après sa résolution (par la méthode du simplexe détaillée au paragraphe suivant) le programme linéaire préliminaire  $\mathbf{I}\mathbf{z} + \mathbf{A}\mathbf{x} = \mathbf{b}$  est transformé en  $\mathbf{T}\mathbf{z} + \mathbf{A}'\mathbf{x} = \mathbf{b}'$ . Trois cas peuvent se présenter :

1. Au moins une des composantes de  $\mathbf{z}$  est non nulle (la matrice  $\mathbf{T}$  contient au moins un vecteur colonne qui est un vecteur de base  $\mathbf{e}_k$  associé à une  $k$ -ième composante de  $\mathbf{b}'$  non nulle). Dans ce cas le programme linéaire  $\mathbf{A}\mathbf{x} = \mathbf{b}$  ne possède pas de solution réalisable (c'est-à-dire avec  $\mathbf{x} > \mathbf{0}$ )



2. Les composantes de  $\mathbf{z}$  sont toutes hors base (la matrice  $\mathbf{T}$  ne contient aucun vecteur colonne qui soit un vecteur de base  $\mathbf{e}_k$ , ils sont tous dans  $\mathbf{A}'$ ).  $\mathbf{z} = \mathbf{0}$  est solution. Le système  $\mathbf{A}'\mathbf{x} = \mathbf{b}'$  est quasiment sous forme simpliciale. On peut réordonner les colonnes de  $\mathbf{A}'$  (et les composantes de  $\mathbf{x}$ ) pour former la matrice identité en tête à partir des  $m$  vecteurs de base  $\mathbf{e}_k$ .
3. Les composantes de  $\mathbf{z}$  qui sont dans la base sont nulles (la matrice  $\mathbf{T}$  contient au moins un ou des vecteurs colonnes de base  $\mathbf{e}_k$ , mais associés à une ou des composante de  $\mathbf{b}'$  nulles).  $\mathbf{z} = \mathbf{0}$  est solution. Mais les vecteurs de base  $\mathbf{e}_k$  qui sont dans  $\mathbf{T}$  font défaut dans  $\mathbf{A}'$ . Il faut donc réaliser des opérations de pivotage pour les faire apparaître dans  $\mathbf{A}'$ . Soit  $z_i$  la composante associée à la colonne  $\mathbf{e}_k$ . On recherche dans la  $k$ -ième ligne de  $\mathbf{A}'$  un élément  $a'_{k,j}$  non nul. S'il n'y en a pas, cela signifie que la matrice  $\mathbf{A}$  n'était pas de rang complet. La ligne peut être supprimée. Soit  $j$  une colonne de  $\mathbf{A}'$  où se trouve un  $a'_{k,j} \neq 0$ . Par une pré-multiplication par  $\mathbf{T}'_{k,j}$  (cf paragraphe 2.5.4 du mémo "Algorithmes numériques pour l'algèbre linéaire") on fait apparaître le vecteur colonne  $\mathbf{e}_k$  en colonne  $j$  de  $\mathbf{A}'$  et la même opération est faite sur  $\mathbf{T}$  et sur  $\mathbf{b}'$  ce qui a pour effet de faire passer de faire passer  $x_j$  dans la base et d'en sortir  $z_i$ . L'opération est renouvelée jusqu'à ce que tous les  $z_i$  soient hors base. On est alors ramené au cas 2.

### 4.2.2 Passage d'un sommet à l'autre

Supposons le programme sous la forme simpliciale suivante :

$$\begin{aligned} \min f &= \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \\ \mathbf{I} \mathbf{x}_B + \mathbf{R} \mathbf{x}_N &= \mathbf{b} \geq \mathbf{0} \end{aligned}$$

D'où :

$$\begin{aligned} \mathbf{x}_B &= \mathbf{b} - \mathbf{R} \mathbf{x}_N \\ f &= (\mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{R}) \mathbf{x}_N + \mathbf{c}_B^T \mathbf{b} \end{aligned}$$

$\mathbf{x}_B = \mathbf{b}$  et  $\mathbf{x}_N = \mathbf{0}$  constitue une solution de base réalisable de critère  $f = \mathbf{c}_B^T \mathbf{b}$ .

Notons :

$$\mathbf{k}^T = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{R}$$

les *coûts réduits* associés aux variables hors base  $\mathbf{x}_N$ . Comme elles sont nulles, elle n'ont que la possibilité d'augmenter. Pour changer de sommet on doit augmenter un seul des  $x_N j$  que l'on nomme *variable entrante* (dans la base) et annuler un des  $x_B i$  que l'on nomme *variable sortante* (de la base) et les permuter dans les sous-ensembles  $\mathbf{x}_B$  et  $\mathbf{x}_N$ . Le nouveau critère varie de la quantité  $k_j x_N j$ . Pour qu'il diminue alors qu'on augmente un  $x_N j$ , il faut que  $k_j$  soit négatif. Si tous les  $k_j$  sont positifs ou nuls la solution actuelle est optimale.

Si  $\mathbf{k} \geq \mathbf{0}$  la solution optimale est trouvée.

Dans le cas contraire, un ou plusieurs  $k_j$  sont négatifs. Soit  $j$  l'indice de l'un d'entre eux. Le fait d'augmenter  $x_N j$  modifie les variables de base  $\mathbf{x}_B = \mathbf{b}$  qui deviennent  $\mathbf{x}_B = \mathbf{b} - \mathbf{r}_j x_N j$  où  $\mathbf{r}_j$  est la  $j$ -ième colonne de  $\mathbf{R}$ . Si  $\mathbf{r}_j \leq \mathbf{0}$ , on peut augmenter  $x_N j$  avec toujours  $\mathbf{x}_B \geq \mathbf{0}$ . Il en résulte que :

Si  $\exists k_j < 0$  tel que  $\mathbf{r}_j \leq \mathbf{0}$  il n'y a pas de minimum fini puisqu'on peut augmenter  $x_N j$  à l'infini. (Remarque, si  $\exists k_j > 0$  tel que  $\mathbf{r}_j \leq \mathbf{0}$ , il n'y a pas de maximum fini pour le critère max  $f = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N$ ).

Dans le cas contraire, on considère uniquement les lignes d'indice  $i$  telles que  $r_{i,j} > 0$ . Les  $x_B i$  correspondants diminuent de la quantité  $r_{i,j} x_N j$  quand  $x_N j$  augmente, ce qui entraîne le passage à zéro de ces  $x_B i$  pour  $x_N j = \frac{b_i}{r_{i,j}}$ . Il faut arrêter l'augmentation de  $x_N j$  au premier  $x_B i$  qui devient nul, d'où :

$$x_N j = \min_i \frac{b_i}{r_{i,j}} \text{ pour } r_{i,j} > 0 \rightarrow x_B i = 0$$

On effectue ensuite l'opération de pivotage qui permet l'échange entre  $x_{Nj}$  et  $x_{Bi}$ . Cette opération comporte deux étapes :

1) Calcul de la nouvelle matrice  $\mathbf{R}$  et du nouveau vecteur  $\mathbf{b}$  en les pré-multipliant par la matrice  $\mathbf{T}_{i,j}^l$  (cf paragraphe 2.5.4 du mémo "Algorithmes numériques pour l'algèbre linéaire"). Dans cette multiplication, le système  $\mathbf{I}\mathbf{x}_B + \mathbf{R}\mathbf{x}_N = \mathbf{b}$  devient  $\mathbf{I}'\mathbf{x}_B + \mathbf{R}'\mathbf{x}_N = \mathbf{b}'$  avec  $\mathbf{I}' = \mathbf{T}_{i,j}^l$ ,  $\mathbf{R}' = \mathbf{T}_{i,j}^l \mathbf{R}$ , et  $\mathbf{b}' = \mathbf{T}_{i,j}^l \mathbf{b}$ . La matrice  $\mathbf{T}_{i,j}^l$  est choisie telle que la  $j$ -ième colonne de la matrice  $\mathbf{R}'$  se transforme en vecteur de base  $\mathbf{e}_i$ . Dans cette transformation  $\mathbf{I}$  reste identité à l'exception de sa  $i$ -ième colonne  $\mathbf{e}_i$  qui devient  $\mathbf{i}'_i$  égal à la  $i$ -ième colonne de  $\mathbf{T}_{i,j}^l$  (c'est-à-dire le vecteur  $-\frac{1}{r_{i,j}}\mathbf{r}_j$  à l'exception de la  $i$ -ième composante qui vaut  $\frac{1}{r_{i,j}}$ ). Cette pré-multiplication par une matrice régulière ne modifie en rien le système.

Plus simplement, dans cette opération, un élément quelconque  $z_k$  de  $\mathbf{I}$ ,  $\mathbf{R}$  ou  $\mathbf{b}$  situé à une ligne d'indice  $k \neq i$  devient  $z'_k = z_k - r_{kj}z_j/r_{ij}$  et un élément  $z_i$  de  $\mathbf{I}$ ,  $\mathbf{R}$  ou  $\mathbf{b}$  situé à une ligne d'indice  $i$  devient  $z'_i = z_i/r_{ij}$ .

2) Echange des vecteurs colonnes  $\mathbf{i}'_i$  de  $\mathbf{I}'$  et  $\mathbf{r}'_j = \mathbf{e}_i$  de  $\mathbf{R}'$  et calcul de  $\mathbf{c}'_N$  et  $\mathbf{c}'_B$  par échange des composantes  $c_{Bi}$  et  $c_{Nj}$  de  $\mathbf{c}_N$  et  $\mathbf{c}_B$ .

Le système est de nouveau sous forme simpliciale. Un nouvel échange de sommet peut être effectué. Ce processus est itéré jusqu'à ce que la solution optimale soit trouvée (signalée par  $\mathbf{k} \geq \mathbf{0}$ ) ou bien jusqu'à ce qu'une condition de minimum non bornée soit trouvée (signalée par  $\exists k_j < 0$  tel que  $\mathbf{r}_j \leq \mathbf{0}$ ).

**Remarque :** Après l'échange de sommet  $\mathbf{I}'$  redevient  $\mathbf{I}$ . Les opérations sur  $\mathbf{I}'$  sont donc inutiles. Par ailleurs, le nouveau vecteur des coûts réduits  $\mathbf{k}' = \mathbf{c}'_N - \mathbf{R}'^T \mathbf{c}'_B$  peut être calculé en ajoutant la ligne supplémentaire  $\mathbf{k}^T$  à la matrice  $\mathbf{R}$  et en lui faisant subir la même opération qu'aux autres lignes  $\mathbf{R}$ . Il en résulte que l'algorithme du simplexe peut être mis en oeuvre en effectuant les opérations de pivotage à la matrice augmentée :

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{b} \\ \mathbf{k}^T & \alpha \end{bmatrix}$$

de dimension  $(m+1) \times (r+1)$  où  $\alpha$  est l'opposé du critère dans le cas d'une minimisation.  $\alpha$  est initialisé par la valeur  $-\mathbf{c}_B^T \mathbf{x}_B$ . La colonne sortante  $j$  (transformée en vecteur  $\mathbf{e}_i$  par  $\mathbf{T}_{i,j}^l$ ) est remplacée par la  $i$ -ième colonne de  $\mathbf{T}_{i,j}^l$ .

### 4.2.3 Règle de sélection de la variable entrante

Lorsque plusieurs  $k_j = c_{Nj} - \mathbf{c}_B^T \mathbf{r}_j$  sont négatifs, on peut choisir pour variable entrante  $x_{Nj}$  :

a) celle qui correspond à l'indice  $J_N(j)$  minimal, relativement aux indices  $j_N = J_N(j)$  des composantes de  $\mathbf{x}_N$  dans le vecteur  $\mathbf{x}$  initial (Début de la règle de Bland<sup>1</sup>),

b) celle qui correspond au  $k_j$  le plus négatif.

c) celle qui conduit à une diminution maximale du coût après échange. Pour tous  $j$  tels que  $k_j < 0$  on calcule la variation de critère produite par le changement de sommet et on choisit la plus grande diminution :

$$\Delta_{i,j} = k_j \left( \min_{i|r_{i,j}>0} \left( \frac{b_i}{r_{i,j}} \right) \right) \rightarrow j = \arg \min_{j|k_j<0} \Delta_{i,j}$$

Les règles les plus employées sont les deux premières (les plus simples) car les tests plus complexes réalisés par la troisième sont des tests locaux qui ne permettent pas d'identifier le chemin globalement le plus court. Un grand gain initial peut très bien amener sur le plus long des itinéraires.

1. R.G. Bland. "New finite pivoting rules for the simplex method". Mathematics of Operations Research, Vol. 2, pages 103-107, 1977.

#### 4.2.4 Dégénérescence

Le premier type de dégénérescence se produit lorsqu'à l'issue d'une itération  $\mathbf{x}_B$  comporte une ou des composantes nulles. Dans ce cas, le processus de passage d'un sommet à l'autre peut se mettre à cycliser. Pendant ce cycle des échanges vont se faire entre des couples de variables  $x_{N_j}$  et  $x_{B_i}$  nulles et qui restent nulles, pour revenir au bout de quelques échanges à la même partition  $(\mathbf{x}_B, \mathbf{x}_N)$ . En fait, il n'y a pas de véritable changement de sommet qui est, dans ce cas, l'intersection d'un nombre de contraintes supérieur à  $r$ . C'est par exemple le cas de l'origine  $\mathbf{0}$  lorsque le vecteur  $\mathbf{b}$  a des composantes nulles.

Une solution qui comporte une ou plusieurs composantes de  $\mathbf{x}_B$  nulles est dite dégénérée.

Le deuxième type de dégénérescence a lieu lorsque la solution est telle que  $\mathbf{k}$  n'est pas strictement positif. Il y a donc une ou des variables hors base qui peuvent augmenter sans que le critère ne soit altéré. Soit  $x_{N_j}$  une de ces variables hors base. Si  $\mathbf{r}_j \leq 0$ , on peut augmenter infiniment  $x_{N_j}$  sans modifier le critère. Le critère est fini, mais, il y a une infinité non bornée de solutions. Sinon, il y a une infinité bornée de solutions (une arête, une face, ...).

#### 4.2.5 Règle d'anti-cyclage

Plusieurs règles ont été proposées pour éviter le phénomène de cyclage. La plus simple est la *règle de Bland*. Elle peut être appliquée en permanence, ou seulement quand il y a risque de dégénérescence, c'est-à-dire lorsque le vecteur  $\mathbf{b}$  a des composantes nulles. Dans ce cas on choisit pour variable entrante  $x_{N_j}$  celle qui correspond au  $J_N(j)$  minimal (indice  $J_N$  relatif à  $\mathbf{x}$  initial), puis parmi les  $r_{i,j} > 0$ , si plusieurs indices  $i$  fournissent le même minimum  $\min_i \frac{b_i}{r_{i,j}}$  (nul à cause du fait que plusieurs  $b_i$  sont nuls), on prend comme variable sortante  $x_{B_i}$  celle qui correspond à l'indice  $I_B(i)$  minimal (indice  $I_B$  relatif à  $\mathbf{x}$  initial).

### 4.3 Passage à la forme standard

Les transformations suivantes sont utilisées pour ramener un programme linéaire quelconque à la forme standard.

#### 4.3.1 Variables négatives

Une variable sujette à la contrainte  $x_j \leq 0$  est remplacée par la variable  $x'_j = -x_j \geq 0$

#### 4.3.2 Variables non contraintes

Une variable  $x_j$  non contrainte est remplacée par la différence de deux variables positives  $x'_j$  et  $x''_j$  :

$$x_j = x'_j - x''_j \text{ avec } \begin{cases} x'_j \geq 0 \\ x''_j \geq 0 \end{cases}$$

#### 4.3.3 Contraintes inégalités

Les contraintes inégalités sont écrites de telle manière que le deuxième membre soit systématiquement positif ou nul, à savoir :  $\mathbf{I}_i^T \mathbf{x} \leq b_i, b_i \geq 0$  et  $\mathbf{I}_k^T \mathbf{x} \geq b_k, b_k \geq 0$ . Chacune de ces contraintes est

transformée en une contrainte égalité par l'ajout d'une variable d'écart positive :

$$\begin{aligned} \mathbf{l}_i^T \mathbf{x} \leq b_i &\rightarrow \mathbf{l}_i^T \mathbf{x} + y_i = b_i \text{ avec } y_i \geq 0 \\ \mathbf{l}_k^T \mathbf{x} \geq b_k &\rightarrow \mathbf{l}_k^T \mathbf{x} - y_k = b_k \text{ avec } y_k \geq 0 \end{aligned}$$

Après ces transformations, le système est sous forme standard, prêt pour une résolution par la méthode du simplexe.

## 4.4 La dualité

La dualité est une correspondance établie entre deux programmes linéaires qui ne mettent en jeu **que des contraintes inégalités** faisant intervenir la matrice  $\mathbf{A}$  dans un programme et  $\mathbf{A}^T$  dans l'autre. Considérons par exemple le programme linéaire *primal* suivant :

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.c.} \quad & \begin{cases} \mathbf{A} \mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} \end{aligned}$$

Prenons  $\mathbf{y} \geq \mathbf{0}$  et  $\boldsymbol{\mu} \geq \mathbf{0}$  comme paramètres de Kuhn et Tucher (cf. paragraphe 3.3) associés aux contraintes inégalités. Le lagrangien s'écrit :

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\mu}) = \mathbf{c}^T \mathbf{x} - \mathbf{y}^T (\mathbf{A} \mathbf{x} - \mathbf{b}) - \boldsymbol{\mu}^T \mathbf{x}$$

La condition de stationnarité s'écrit :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{x}} &= \mathbf{c} - \mathbf{A}^T \mathbf{y} - \boldsymbol{\mu} = \mathbf{0} \\ \boldsymbol{\mu} &\geq \mathbf{0} \rightarrow \mathbf{A}^T \mathbf{y} \leq \mathbf{c} \end{aligned}$$

Par ailleurs, remarquons que si  $\mathbf{x}$  et  $\mathbf{y}$  sont admissibles, on a :

$$\mathbf{c} \geq \mathbf{A}^T \mathbf{y} \rightarrow \mathbf{c}^T \mathbf{x} \geq \mathbf{y}^T \mathbf{A} \mathbf{x} \geq \mathbf{y}^T \mathbf{b} \rightarrow \mathbf{c}^T \mathbf{x} \geq \mathbf{b}^T \mathbf{y}$$

Considérons le problème *dual* suivant :

$$\begin{aligned} \max_{\mathbf{y}} \quad & \mathbf{b}^T \mathbf{y} \\ \text{s.c.} \quad & \begin{cases} \mathbf{A}^T \mathbf{y} \leq \mathbf{c} \\ \mathbf{y} \geq 0 \end{cases} \end{aligned}$$

Prenons  $\mathbf{x} \geq \mathbf{0}$  et  $\mathbf{v} \geq \mathbf{0}$  comme paramètres de Kuhn et Tucker associés aux contraintes inégalités. Le lagrangien s'écrit :

$$\mathcal{L}'(\mathbf{y}, \mathbf{x}, \mathbf{v}) = \mathbf{b}^T \mathbf{y} - \mathbf{x}^T (\mathbf{A}^T \mathbf{y} - \mathbf{c}) + \mathbf{v}^T \mathbf{y}$$

La condition de stationnarité s'écrit :

$$\begin{aligned} \frac{\partial \mathcal{L}'}{\partial \mathbf{y}} &= \mathbf{b} - \mathbf{A} \mathbf{x} + \mathbf{v} = \mathbf{0} \\ \mathbf{v} &\geq \mathbf{0} \rightarrow \mathbf{A} \mathbf{x} \geq \mathbf{b} \end{aligned}$$

On a ainsi deux programmes dont les variables de l'un sont les paramètres de Kuhn et Tucker de l'autre et réciproquement.

Si on compare les lagrangiens, on constate que :

$$\mathcal{L}'(\mathbf{y}, \mathbf{x}, \nu) - \nu^T \mathbf{y} = \mathcal{L}(\mathbf{x}, \mathbf{y}, \mu) + \mu^T \mathbf{x}$$

Or  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mu$  et  $\nu$  doivent vérifier  $\nu^T \mathbf{y} \geq \mathbf{0}$  et  $\mu^T \mathbf{x} \geq \mathbf{0}$ , d'où :

$$\mathcal{L}'(\mathbf{y}, \mathbf{x}, \nu) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}, \mu)$$

Par contre à l'optimum, s'ils existent, on a  $\hat{\nu}^T \hat{\mathbf{y}} = 0$  et  $\hat{\mu}^T \hat{\mathbf{x}} = 0$ , d'où :

$$\mathcal{L}'(\hat{\mathbf{y}}, \hat{\mathbf{x}}, \hat{\nu}) = \mathcal{L}(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mu})$$

et compte tenu du fait que  $\hat{\mathbf{x}}^T (\mathbf{A}^T \hat{\mathbf{y}} - \mathbf{c}) = 0$  et  $\hat{\mathbf{y}}^T (\mathbf{A} \hat{\mathbf{x}} - \mathbf{b}) = 0$ , il en résulte que :

$$\mathbf{c}^T \hat{\mathbf{x}} = \mathbf{b}^T \hat{\mathbf{y}}$$

### En résumé :

Les programmes linéaires :

$$\begin{array}{cc} \text{primal} & \text{dual} \\ \min_{\mathbf{x}} \mathbf{c}^T \mathbf{x} & \max_{\mathbf{y}} \mathbf{b}^T \mathbf{y} \\ \text{s.c.} \begin{cases} \mathbf{A} \mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} & \text{s.c.} \begin{cases} \mathbf{A}^T \mathbf{y} \leq \mathbf{c} \\ \mathbf{y} \geq 0 \end{cases} \end{array}$$

1. ont des solutions admissibles  $\mathbf{x}$  et  $\mathbf{y}$  telles que  $\mathbf{c}^T \mathbf{x} \geq \mathbf{b}^T \mathbf{y}$
2. ont des solutions optimales  $\hat{\mathbf{x}}$  et  $\hat{\mathbf{y}}$  telles que  $\mathbf{c}^T \hat{\mathbf{x}} = \mathbf{b}^T \hat{\mathbf{y}}$
3. les paramètres de Kuhn et Tucker d'un programme sont les variables de l'autre et réciproquement,
4. si un programme n'est pas borné son dual est vide (pas de solution admissible). Si un programme n'a pas de solution admissible, son dual est soit non borné, soit vide.

### Remarque :

En changeant les signes de  $\mathbf{A}$ ,  $\mathbf{b}$ , ou  $\mathbf{c}$  on peut donner d'autres formes à cette dualité. On peut changer une minimisation en maximisation et inversement en changeant le signe de  $\mathbf{c}$ . Si le problème à traiter comporte des contraintes inégalités des deux sens, on peut les ramener toutes dans le même en changeant de signe les lignes concernées. Les contraintes égalités du type  $\mathbf{I}_i^T \mathbf{x} = b_i$  sont traitées comme deux contraintes inégalités,  $\mathbf{I}_i^T \mathbf{x} \leq b_i$  et  $-\mathbf{I}_i^T \mathbf{x} \leq -b_i$  dans le cas standard de la maximisation, ou  $\mathbf{I}_i^T \mathbf{x} \geq b_i$  et  $-\mathbf{I}_i^T \mathbf{x} \geq -b_i$  dans le cas standard de la minimisation. Les paramètres de Kuhn et Tucker (positifs) associés à ces deux contraintes sont deux variables du dual dont la différence (de signe quelconque) représente la variable de Lagrange associée à la contrainte égalité  $\mathbf{I}_i^T \mathbf{x} = b_i$ .

### Utilisation :

Supposons que  $\mathbf{A}$  soit de dimension  $m \times r$ , que les variables  $\mathbf{X}^T = (\mathbf{x}^T \quad \nu^T)$  du primal aient été indicées de 1 à  $n = m + r$  et que l'on ait la solution du *primal*, sous la forme de la matrice  $\mathbf{M}'_p$  finale :

$$\mathbf{M}'_p = \begin{bmatrix} \mathbf{R}'_p & \mathbf{b}' \\ \mathbf{k}'_p & \alpha'_p \end{bmatrix}$$

avec  $\mathbf{k}'_p \geq 0$  (solution atteinte). Supposons que la liste des indices des variables de base soit donnée par  $I_B(k)$  pour  $k = 1$  à  $m$  et la liste des variables hors base donnée par  $J_N(k)$  pour  $k = 1$  à  $r$ . On a alors :

— Revenu optimal du primal :  $\alpha'_p$  si maximisation ou  $-\alpha'_p$  si minimisation.

- Pour  $k = 1$  à  $m$  :  $i = I_B(k)$ ,  $x_i = b'_k$  si  $i \leq r$  et  $v_i = b'_k$  si  $i > r$  (c'est-à-dire  $\mathbf{I}_{i-r}^T \mathbf{x} - b'_k = b_{i-r}$ ).
- Pour  $k = 1$  à  $r$  :  $j = J_N(k)$ ,  $x_j = 0$  si  $j \leq r$  et  $v_j = 0$  si  $j > r$  (c'est-à-dire  $\mathbf{I}_{j-r}^T \mathbf{x} = b_{j-r}$ ).

Notons  $\mathbf{M}'_D$  la matrice finale qui correspond à la solution du dual, telle que :

$$\mathbf{M}'_D = \begin{bmatrix} \mathbf{R}'_D & \mathbf{c}' \\ \mathbf{k}'_D & \alpha'_D \end{bmatrix}$$

Il est inutile de la calculer car, si on indice les variables du dual dans l'ordre  $\mathbf{Y}^T = (\boldsymbol{\mu}^T \quad \mathbf{y}^T)$  de 1 à  $n$ , les éléments de  $\mathbf{M}'_D$  figurent tous dans  $\mathbf{M}'_P$ . En effet, on a :

$$\mathbf{M}'_P = \begin{bmatrix} \mathbf{R}'_P & \mathbf{k}'_P \\ \mathbf{b}'^T & \alpha'_P \end{bmatrix} = \begin{bmatrix} -\mathbf{R}'_D & \mathbf{c}' \\ \mathbf{k}'_D & \alpha'_D \end{bmatrix}$$

Il en résulte que :

- Revenu optimal du dual :  $\alpha'_P$  si maximisation ou  $-\alpha'_P$  si minimisation.
- Pour  $k = 1$  à  $m$  :  $i = I_B(k)$ ,  $\mu_i = 0$  si  $i \leq r$  et  $y_i = 0$  si  $i > r$ .
- Pour  $k = 1$  à  $r$  :  $j = J_N(k)$ ,  $\mu_j = k'_{Pk}$  si  $j \leq r$  et  $y_j = k'_{Pk}$  si  $j > r$ .

où tous les résultats, y compris les listes  $I_B(k)$  et  $J_N(k)$ , proviennent de la résolution du primal.

**Conclusion :** Lorsqu'un problème se présente, si on ne dispose pas d'une solution de base réalisable, on peut regarder si le dual en possède une. On peut ainsi éviter d'effectuer un simplexe en deux phases. La question de choix, entre les deux résolutions se pose quand primal et dual possèdent tous deux une solution de base réalisable. Généralement, on considère qu'il est plus rapide de résoudre le programme qui possède le moins de lignes (puisque le nombre total d'inconnues, en comptant les variables d'écart est le même).

## Chapitre 5

# Optimisation positive

Nous appelons optimisation positive l'étude d'un problème pour lequel les paramètres inconnus doivent tous être de même signe (positifs par exemple). Ces paramètres sont a priori surabondants. Une optimisation est effectuée pour choisir la meilleure solution. Ce problème se présente par exemple lorsqu'on recherche avec quelles tensions  $\mathbf{x}$  il faut tirer sur  $\dim(\mathbf{x}) = n \geq 7$  câbles pour imposer un torseur d'effort donné à un solide attaché par ces  $n$  câbles, par exemple pour compenser son poids et le tenir en équilibre, ou pour lui communiquer une accélération désirée ce qui se traduit par une relation linéaire  $\mathbf{Ax} = \mathbf{b}$  (avec dans ce cas  $\dim(\mathbf{b}) = m = 6$  composantes, 3 en force et 3 en moment). Tous les câbles doivent être tendus pour que la suspension de l'objet soit opérationnelle (contraintes inégalités  $\mathbf{x} \geq 0$ ). On se pose le problème de l'existence de solutions et du choix de la meilleure, lorsqu'il en existe.

### 5.1 Enoncé du problème

Cherchons dans quelles conditions le problème :

$$\begin{cases} \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} \quad (5.1)$$

a des solutions.

Notons  $\mathbf{a}_j$  les vecteurs colonnes de  $\mathbf{A}$  et  $x_j$  les composantes de  $\mathbf{x}$ .

L'égalité  $\mathbf{b} = \sum_{j=1}^n x_j \mathbf{a}_j$  montre que si le système 5.1 a une solution, alors  $\mathbf{b}$  est à l'intérieur de l'hypercône d'origine  $\mathbf{0}$  tendu par les vecteurs  $\mathbf{a}_j$ , car c'est une combinaison linéaire à coefficients positifs de ces vecteurs.

### 5.2 Théorèmes de l'hyperplan séparateur

Considérons l'existence d'un vecteur  $\mathbf{u}$  de  $\mathbb{R}^m$  non nul tel que  $\mathbf{u}^T \mathbf{A} \geq \mathbf{0}$ . Ce vecteur fait un angle non obtus avec tous les vecteurs  $\mathbf{a}_j$  qui sont d'un même coté par rapport à l'hyperplan normal à  $\mathbf{u}$ . On a alors :

$$\mathbf{u}^T \mathbf{Ax} \geq 0$$

puisque tous les  $x_j$  sont positifs. Il en résulte que  $\mathbf{u}^T \mathbf{b} \geq 0$ . En conséquence :

$$\text{Si } \begin{cases} \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} \text{ a une solution, il n'existe pas } \mathbf{u} \text{ de } \mathbb{R}^m \text{ tel que } \mathbf{u}^T \mathbf{A} \geq \mathbf{0} \text{ et } \mathbf{u}^T \mathbf{b} < 0.$$

Autrement dit, si notre système a une solution, il n'existe pas d'hyperplan séparateur (normal à un certain vecteur  $\mathbf{u}$ ) tel que tous les  $\mathbf{a}_j$  soient d'un côté et le vecteur  $\mathbf{b}$  de l'autre.

Remarque : Le côté des  $\mathbf{a}_j$  inclut l'hyperplan, le côté de  $\mathbf{b}$  l'exclut.

Supposons maintenant que  $\mathbf{Ax} = \mathbf{b}$  n'a pas de solution. Dans ce cas  $\mathbf{b}$  peut être décomposé en une somme de deux vecteurs orthogonaux  $\mathbf{b} = \mathbf{b}_e + \mathbf{b}_n$  avec  $\mathbf{b}_n \neq \mathbf{0}$  tels qu'il existe  $\mathbf{x}$  vérifiant  $\mathbf{Ax} = \mathbf{b}_e$  et  $\mathbf{b}_n^T \mathbf{A} = \mathbf{0}$ . D'où en posant  $\mathbf{u} = -\mathbf{b}_n$ , on a  $\mathbf{u}^T \mathbf{b} = -\mathbf{b}_n^T \mathbf{b}_n = -\|\mathbf{b}_n\|^2 < 0$ .

La démonstration de cette réciproque dans le cas où  $\mathbf{Ax} = \mathbf{b}$  a une solution, mais systématiquement avec une ou plusieurs composantes  $x_j$  négatives est plus complexe (cf la démonstration par récurrence de Gale<sup>1</sup>).

En conséquence :

Si  $\begin{cases} \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases}$  n'a pas de solution, il existe  $\mathbf{u}$  de  $\mathbb{R}^m$  tel que  $\mathbf{u}^T \mathbf{A} \geq \mathbf{0}$  et  $\mathbf{u}^T \mathbf{b} < 0$ .

Autrement dit, si notre système n'a pas de solution, il existe un hyperplan séparateur (normal à un certain vecteur  $\mathbf{u}$ ) tel que tous les  $\mathbf{a}_j$  sont d'un côté et le vecteur  $\mathbf{b}$  de l'autre.

### 5.3 Le théorème de Farkas

Si pour tout  $\mathbf{u}$  tel que  $\mathbf{u}^T \mathbf{A} \geq \mathbf{0}$  on a  $\mathbf{u}^T \mathbf{b} \geq 0$ , alors le système  $\mathbf{Ax} = \mathbf{b}$  admet une solution  $\mathbf{x} \geq \mathbf{0}$ .

A défaut d'une démonstration, donnons une interprétation physique à ce théorème. Il signifie que les  $n$  vecteurs  $\mathbf{a}_j$  (les colonnes) forment un hyper-cône pyramidal d'origine  $\mathbf{0}$  dont les demi-angles au sommet entre les arêtes  $\mathbf{a}_j$  ne sont pas obtus car cet hyper-cône est toujours entièrement d'un même côté des l'hyperplans normaux aux vecteurs  $\mathbf{u}$  tels que  $\mathbf{u}^T \mathbf{A} \geq \mathbf{0}$ . Comme pour chacun des ces vecteurs  $\mathbf{u}$ , on a  $\mathbf{u}^T \mathbf{b} \geq 0$ ,  $\mathbf{b}$  est également du même côté de l'hyperplan. En choisissant les hyperplans d'appui sur l'enveloppe convexe des points  $\mathbf{a}_j$  et du point  $\mathbf{0}$  origine, on constate que  $\mathbf{b}$  est intérieur à cette enveloppe convexe. Il en résulte qu'on peut trouver une combinaison linéaire des  $\mathbf{a}_j$  à coefficients positifs pour réaliser le vecteur  $\mathbf{b}$ .

### 5.4 Matrices opérationnelles et noyau coopératif

Cherchons dans quelles conditions la problème :

$$\forall \mathbf{b}, \exists \mathbf{x} > \mathbf{0} \mid \mathbf{Ax} = \mathbf{b} \quad (5.2)$$

a des solutions. Plus simplement, on cherche à caractériser les matrices  $\mathbf{A}$  qui permettent d'obtenir des solutions  $\mathbf{x}$  strictement positives quelle que soit la valeur du second membre  $\mathbf{b}$ . Généralement, la matrice  $\mathbf{A}$  représente un processus, les caractéristiques obtenues indiquerons comment agir sur le processus pour qu'il soit apte à fournir des solutions positives  $\mathbf{x}$  dans tous les cas.

En premier lieu, il est évident que pour obtenir une solution  $\forall \mathbf{b}$ , la matrice  $\mathbf{A}$  doit être de rang complet ( $\text{rang}(\mathbf{A}) = m = \dim(\mathbf{b})$ ). Mais cette condition ne suffit pas, à elle seule. Pour caractériser la propriété supplémentaire qui la rend apte à réaliser l'opération demandée, nous avons introduit la notion de matrice opérationnelle. Généralement la matrice  $\mathbf{A}$  dépend de paramètres  $\mathbf{p}$  par une relation  $\mathbf{A}(\mathbf{p})$ . Ces paramètres évoluent dans un espace que l'étude de  $\mathbf{A}$  permet de scinder en région régulière (où  $\text{rang}(\mathbf{A}) = m$ ) et en région singulière (où  $\text{rang}(\mathbf{A}) < m$ ). Mais pour le problème traité, la région régulière doit encore être scindée en région opérationnelle et en région inopérationnelle.

1. D. Gale. "The basic theorems of real linear equations, inequalities, linear programming and game theory", Naval Research Logistics Quarterly, Vol. 3 (1956) pages 193-200.



En second lieu, on s'aperçoit immédiatement que si la matrice  $\mathbf{A}$  est carrée, elle ne peut être opérationnelle. En effet, si  $n = m = \text{rang}(\mathbf{A})$ , la solution  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$  est unique pour un  $\mathbf{b}$  donné. Si  $\mathbf{x} > \mathbf{0}$ , le système  $\mathbf{Ax} = \mathbf{b}'$  avec  $\mathbf{b}' = -\mathbf{b}$  ne peut avoir de solution opérationnelle.

*Une matrice opérationnelle est forcément redondante et de rang complet.*

Notons  $\mathbf{l}_i, i = 1, m$  les lignes (qui ont  $n$  composantes) de la matrice  $\mathbf{A}$  et  $\mathbf{a}_j, j = 1$  à  $n$  ses colonnes (qui ont  $n$  composantes).

#### 5.4.1 Cas de la redondance simple : $n = m + 1$

Dans ce cas, le noyau  $\mathbf{n}$  (qui a  $n$  composantes) de la matrice  $\mathbf{A}$  a une direction unique donné par :

$$\mathbf{n} = \mathbf{l}_1 \wedge \mathbf{l}_2 \wedge \dots \wedge \mathbf{l}_m$$

(les composantes de  $\mathbf{n}$  sont les cofacteurs de la première ligne de la matrice  $n \times n$  dont les  $n - 1$  lignes suivantes sont les  $\mathbf{l}_i$ ). Notons  $\mathbf{A}_j$  la matrice carrée  $m \times m$  où on a supprimé la colonne d'indice  $j$ . La  $j$ -ème composante de  $\mathbf{n}$  est donnée par :

$$n_j = (-1)^j \det(\mathbf{A}_j)$$

La solution générale au système  $\mathbf{Ax} = \mathbf{b}$  peut s'écrire :

$$\mathbf{x} = \mathbf{x}^{eff} + \mathbf{x}^{nul} \text{ avec } \mathbf{x}^{nul} = \lambda \mathbf{n}$$

où  $\mathbf{x}^{eff}$  est par exemple la solution "efficace" de norme minimale calculé au moyen de la pseudo-inverse de Moore-Penrose :

$$\mathbf{x}^{eff} = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{b}$$

Comme  $\mathbf{x}^{eff}$  peut prendre n'importe quelle valeur, et en particulier  $\mathbf{0}$  lorsque  $\mathbf{b} = \mathbf{0}$ , pour que  $\mathbf{x}$  soit strictement positif, il faut que  $\mathbf{n} > \mathbf{0}$ . Réciproquement, si  $\mathbf{n} > \mathbf{0}$ , on voit qu'avec un  $\lambda$  suffisamment grand on peut rendre  $\mathbf{x} > \mathbf{0}$  quel que soit  $\mathbf{x}^{eff}$ .

**Remarque :** Tout vecteur du noyau est défini à un coefficient (positif ou négatif) multiplicatif près. Dire que  $\mathbf{n} > \mathbf{0}$  signifie en fait qu'il a toutes ses composantes de même signe.

**Définition : noyau coopératif.**

Nous dirons que le noyau de  $\mathbf{A}$  est coopératif s'il a toutes ses composantes de même signe.

**Théorème de coopération :**

*Une matrice  $\mathbf{A}$  est opérationnelle, si elle est de rang complet, à noyau coopératif.*

**Le simplexe des  $\mathbf{a}_j$**

La relation  $\mathbf{An} = \sum_{j=1}^n \mathbf{a}_j n_j = \mathbf{0}$  avec  $n_j > 0$  montre que le point origine  $\mathbf{0}$  est barycentre des  $m + 1$  points  $\mathbf{a}_j$  affectés des coefficients positifs  $n_j$ , ce qui signifie que l'origine  $\mathbf{0}$  est à l'intérieur du simplexe formé par les points  $\mathbf{a}_j$ . Réciproquement si  $\mathbf{0}$  est à l'intérieur du simplexe formé par les points  $\mathbf{a}_j$ , on peut trouver  $m + 1$  coefficients  $n_j > 0$  tels que  $\sum_{j=1}^n \mathbf{a}_j n_j = \mathbf{0}$ .

Lorsque le noyau est coopératif, l'origine  $\mathbf{0}$  est à l'intérieur du simplexe formé par les points  $\mathbf{a}_j$  et réciproquement.

Cette propriété est très intéressante pour aider à la construction d'une matrice  $\mathbf{A}$  opérationnelle. En particulier dans le cas  $m = 3$ , on voit tout de suite comment modifier la position de n'importe quel des 4 points  $\mathbf{a}_j$  pour que l'origine  $\mathbf{0}$  rentre à l'intérieur du simplexe (un tétraèdre dans ce cas) qu'ils forment.

Le volume du simplexe est donné par  $V_S = \frac{1}{m!} \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ \mathbf{a}_0 & \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{vmatrix}$ , soit  $V_S = \frac{1}{m!} \sum_{j=1}^{m+1} (-1)^j \det(\mathbf{A}_j)$

où encore  $V_S = \frac{1}{m!} \sum_{j=1}^{m+1} n_j$ .

Si on remplace un des  $\mathbf{a}_j$  par  $\mathbf{0}$  on obtient le volume du simplexe formé par l'origine et les  $m$  points  $\mathbf{a}_k$ ,  $k = 1$  à  $m+1$ , et  $k \neq j$ . En développant ce déterminant selon sa  $j$ -ième colonne on trouve qu'il a pour volume  $V_{Sj} = \frac{1}{m!} (-1)^j \det(\mathbf{A}_j) = \frac{1}{m!} n_j$ . D'où  $V_S = \sum_{j=1}^{m+1} V_{Sj}$ . Si, et seulement si  $\mathbf{0}$  est à l'intérieur du simplexe la valeur absolue du volume du simplexe originel est la somme des valeurs absolues des volumes des  $n = m+1$  simplexes  $m+1$  d'origine  $\mathbf{0}$ , soit  $|V_S| = \sum_{j=1}^{m+1} |V_{Sj}|$ , soit  $\left| \sum_{j=1}^{m+1} n_j \right| = \sum_{j=1}^{m+1} |n_j|$  ce qui n'est possible que si tous les  $n_j$  sont de même signe. On retrouve la condition de coopération du noyau.

### Pas d'hyperplan séparateur

Le fait que l'origine  $\mathbf{0}$  soit l'intérieur du simplexe formé par les points  $\mathbf{a}_j$  implique qu'il ne peut exister d'hyperplan séparateur tel que tous les  $\mathbf{a}_j$  soient d'un même coté de cet hyperplan. En effet, si un tel hyperplan existait, (notons  $\mathbf{u}$  son vecteur normal) on aurait  $\mathbf{u}^T \mathbf{A} > \mathbf{0}^T$ . Le produit scalaire de ce vecteur (qui a toutes ses composantes positives) avec  $\mathbf{n}$  (qui a également toutes ses composantes positives) est forcément positif, d'où  $\mathbf{u}^T \mathbf{A} \mathbf{n} > 0$ , ce qui est en contradiction avec le fait que  $\mathbf{A} \mathbf{n} = \mathbf{0}$ .

### Résolution

La résolution de  $\mathbf{A} \mathbf{x} = \mathbf{b}$  avec  $\mathbf{x} \geq 0$  est immédiate en redondance simple lorsque le noyau est coopératif. On calcule une solution admissible de signe quelconque, par exemple  $\mathbf{x}^{eff} = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{b}$ , puis on calcule le noyau  $\mathbf{n}$  par exemple par  $n_j = (-1)^j \det(\mathbf{A}_j)$  (on change éventuellement tous les signes des  $n_j$  pour qu'ils soient positifs). On calcule :

$$y_{\min} = - \min_{j=1, m+1} \frac{x_j^{eff}}{n_j} \quad (5.3)$$

D'où les solutions :

$$\mathbf{x} = \mathbf{x}^{eff} + y \mathbf{n}, \forall y > y_{\min}$$

Pour  $y = y_{\min}$ , on a la solution de norme minimale qui est telle que  $x_{j_{\min}} = 0$ , où  $j_{\min}$  est l'indice qui correspond à  $y_{\min}$ .

## 5.4.2 Cas de la redondance multiple

Dans le cas d'une redondance multiple le noyau orthonormé  $\mathbf{N}$  de  $\mathbf{A}$  (supposée de rang complet) possède  $r = n - m \geq 2$  directions indépendantes que l'on peut calculer par une factorisation QRE (voir paragraphe 3.7 du mémo "Algorithmes numériques pour l'algèbre linéaire").

La solution générale au système  $\mathbf{A} \mathbf{x} = \mathbf{b}$  peut s'écrire :

$$\mathbf{x} = \mathbf{x}^{eff} + \mathbf{x}^{nul} \text{ avec } \mathbf{x}^{nul} = \mathbf{N} \mathbf{y}, \quad \mathbf{y} \in \mathbb{R}^{n-m} \quad (5.4)$$

Notons  $\mathbf{A}_p$  une sous matrice extraite de la matrice  $\mathbf{A}$  comportant  $p$  colonnes de  $\mathbf{A}$  prises au hasard, avec  $m < p < n$ , et notons  $\mathbf{A}_{n-p}$  la sous matrice comportant les  $n - p$  colonnes restantes. Partitionnons  $\mathbf{x}^{nul}$  en deux sous vecteurs,  $\mathbf{x}_p^{nul}$  à  $p$  composantes et  $\mathbf{x}_{n-p}^{nul}$  à  $n - p$  composantes, tels que :

$$\mathbf{A} \mathbf{x}^{nul} = \mathbf{A}_p \mathbf{x}_p^{nul} + \mathbf{A}_{n-p} \mathbf{x}_{n-p}^{nul} = \mathbf{0}$$

Si la sous matrice  $\mathbf{A}_p$  est opérationnelle, il résulte du théorème de coopération qu'on peut trouver un vecteur  $\mathbf{x}_p^{nul}$  avec toutes ses composantes positives, tel que  $\mathbf{A}_p \mathbf{x}_p^{nul} = -\mathbf{A}_{n-p} \mathbf{x}_{n-p}^{nul}$ , quel que soit le

vecteur  $\mathbf{x}_{n-p}^{nul}$ . En choisissant toutes les composantes de  $\mathbf{x}_{n-p}^{nul}$  positives on génère ainsi un vecteur solution  $\mathbf{x}^{nul}$  coopératif. On en déduit le théorème :

*En redondance multiple, pour qu'une situation soit opérationnelle, il suffit qu'une sous-matrice de rang complet constituée de colonnes de  $\mathbf{A}$  prises au hasard, soit elle-même opérationnelle.*

On peut tester les  $C_n^{m+1} = C_n^{n-m-1}$  sous-matrices  $A_{m+1}$  à redondance simple extraites de  $\mathbf{A}$ . Dès qu'on trouve un sous-noyau coopératif, on peut dire que la matrice est opérationnelle. Le test des  $C_n^{m+1}$  sous-noyaux (36 pour  $n = 6$  et  $m = 9$ ) est nécessaire avant de pouvoir affirmer le contraire. On a ainsi, une procédure systématique qui donne un majorant des calculs à effectuer pour conclure.

#### Relation avec l'enveloppe convexe des $\mathbf{a}_j$

Si l'origine  $\mathbf{0}$  est à l'intérieur d'un des  $C_n^{m+1}$  simplexes différents qui sont réalisables avec les  $n$  points  $\mathbf{a}_j$ , la matrice  $\mathbf{A}$  est opérationnelle. Inversement si l'origine est à l'extérieur des  $C_n^{m+1}$  simplexes, (et en conséquence à l'extérieur de l'enveloppe convexe des  $n$  points  $\mathbf{a}_j$ ), la matrice  $\mathbf{A}$  n'est pas opérationnelle. Par ailleurs, si  $\mathbf{A}$  est de rang complet, un des ces simplexes a un volume non nul et réciproquement. On a ainsi le théorème :

*En redondance multiple, pour qu'une situation soit opérationnelle, il faut et il suffit que l'origine des coordonnées soit à l'intérieur de l'enveloppe convexe du polyèdre à volume non nul formé par les points ayant pour coordonnées les colonnes de la matrice  $\mathbf{A}$ .*

Ce théorème est strictement équivalent à celui que l'on peut déduire du théorème de l'hyperplan séparateur. S'il existe un hyperplan séparateur avec tous les  $\mathbf{a}_j$  d'un même côté, on ne pourra pas trouver de solution quel que soit le vecteur  $\mathbf{b}$ , par contre si un tel hyperplan n'existe pas, on pourra trouver une solution quel que soit  $\mathbf{b}$ .

#### Remarque :

Si une ligne  $\mathbf{l}_i$  de  $\mathbf{A}$  a toutes ses composantes de même signe (aucune nulle), on ne peut avoir  $\mathbf{l}_i^T \mathbf{n} = 0$ , avec un vecteur  $\mathbf{n}$  qui a lui aussi toutes ses composantes positives. Il en résulte qu'il ne peut y avoir de noyau coopératif et que la matrice  $\mathbf{A}$  ne peut être opérationnelle. Malheureusement la réciproque n'est pas vraie, et même s'il n'existe aucune ligne de ce type, on n'est sûr que  $\mathbf{A}$  soit opérationnelle.

## 5.5 Résolution en redondance multiple

Trouver une solution  $\mathbf{x} > \mathbf{0}$  à  $\mathbf{Ax} = \mathbf{b}$  quel que soit  $\mathbf{b}$  revient à trouver une solution  $\mathbf{y} \in \mathbb{R}^{n-m}$  au système :

$$\mathbf{x}^{eff} + \mathbf{Ny} > \mathbf{0}$$

soit :

$$\mathbf{A}^\dagger \mathbf{b} + \mathbf{Ny} > \mathbf{0}$$

où  $\mathbf{A}^\dagger$  est la pseudo-inverse de Moore-Penrose de  $\mathbf{A}$ .

Cette approche est intéressante quand  $r = n - m \ll n$ . C'est elle qui nous a fourni la solution (5.3) dans le cas  $r = 1$ . On peut démontrer par la programmation linéaire, que la solution qui minimise  $\sum x_j$  a  $r$  composantes nulles. On peut chercher la solution  $\mathbf{y}$  d'un de ces  $C_n^r$  systèmes à  $r$  égalités à zéro et voir si les  $m$  autres inégalités sont vérifiées. Cette exploration systématique nécessite au pire la résolution de  $C_n^r$  systèmes d'ordre  $r$ , ce qui n'est intéressant que pour  $r$  et  $n$  très petits. Pour les grandes valeurs de  $r$  et  $n$ , on a une explosion combinatoire de type exponentiel.

### 5.5.1 Critère linéaire - Résolution par le simplexe

En ajoutant un *critère linéaire* à minimiser, par exemple  $\sum x_j$ , on convertit notre problème en programmation linéaire qu'on sait résoudre par la méthode du simplexe.

### 5.5.2 Critère non-linéaire - Méthode de point intérieur

Par contre, si on connaît une solution admissible  $\mathbf{x}^a$  (telle que  $\mathbf{x}^a > \mathbf{0}$  et  $\mathbf{A}\mathbf{x}^a = \mathbf{b}$ ) et qu'on cherche une meilleure solution qui minimise un critère quelconque  $r(\mathbf{x})$  on peut tenter d'améliorer cette solution de la quantité  $k\mathbf{n}$ , avec  $k > 0$ , telle que  $r(\mathbf{x}^a + k\mathbf{n}) < r(\mathbf{x}^a)$  et telle que  $\mathbf{A}(\mathbf{x}^a + k\mathbf{n}) = \mathbf{b}$ . Il en résulte que la direction de  $\mathbf{n}$  doit être telle que :

$$\begin{aligned}\mathbf{A}\mathbf{n} &= \mathbf{0} \\ r_{\mathbf{x}}^T \mathbf{n} &< \mathbf{0}\end{aligned}$$

Cherchons  $\|\mathbf{n}\| = 1$  qui maximise  $-r_{\mathbf{x}}^T \mathbf{n}$ . Le Lagrangien s'écrit :

$$\mathcal{L} = -r_{\mathbf{x}}^T \mathbf{n} + \lambda^T \mathbf{A}\mathbf{n} + \mu (1 - \mathbf{n}^T \mathbf{n})$$

La condition de stationnarité donne :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{n}} = -r_{\mathbf{x}} + \mathbf{A}^T \lambda - \mu \mathbf{n} = \mathbf{0} \rightarrow \mu \mathbf{n} = \mathbf{A}^T \lambda - r_{\mathbf{x}}$$

soit en reportant dans la contrainte  $\mathbf{A}\mathbf{n} = \mathbf{0}$  :

$$\mu \mathbf{A}\mathbf{n} = \mathbf{A}\mathbf{A}^T \lambda - \mathbf{A}r_{\mathbf{x}} = \mathbf{0} \rightarrow \lambda = (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}r_{\mathbf{x}}$$

Il vient :

$$\begin{aligned}\mu \mathbf{n} &= -r_{\mathbf{x}} + \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}r_{\mathbf{x}} \\ &= -\left(\mathbf{I} - \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}\right) r_{\mathbf{x}} \\ &= -\mathbf{N}\mathbf{N}^T r_{\mathbf{x}}\end{aligned}$$

D'où finalement :

$$\mathbf{n} = -\frac{1}{\|\mathbf{N}\mathbf{N}^T r_{\mathbf{x}}\|} \mathbf{N}\mathbf{N}^T r_{\mathbf{x}}$$

La meilleure direction de  $\mathbf{u}$  est celle de la projection orthogonale du gradient du critère sur le noyau de la matrice  $\mathbf{A}$ . Il reste à déterminer la taille du pas à effectuer pour que  $\mathbf{x}^+ = \mathbf{x}^a + k\mathbf{n}$  ne sorte pas du domaine admissible, à savoir  $\mathbf{x}^+ \geq \mathbf{0}$ . Seules les composantes  $n_i < 0$  peuvent rendre  $x_i^+ \leq 0$ . Il en résulte que  $k$  doit être inférieur à  $k_{\max}$  donné par :

$$k_{\max} < \min_{n_i < 0} \frac{-x_i^a}{n_i}$$

Cette *méthode de point intérieur*, plutôt destinée au cas de critères non linéaires, est également utilisée en linéaire comme alternative à de la méthode du simplexe, avec  $k = 0.995k_{\max}$ . Elle présente l'inconvénient de nécessiter un point initial réalisable, ce qui peut être contourné par une recherche préliminaire comme celle employée dans la méthode des deux phases du simplexe.

Nous présentons au chapitre suivant la méthode que nous avons utilisée dans une application réelle pour le calcul des 9 tensions optimales d'une suspension active à 6 degrés de liberté.

## Chapitre 6

# Programmation quadratique

Nous nous limitons notre incursion dans le domaine de la programmation quadratique à la résolution du problème suivant : Trouver le vecteur  $\mathbf{x} \in \mathbb{R}^n$  soumis aux  $m < n$  contraintes linéaires :

$$\mathbf{Ax} = \mathbf{b} \quad (6.1)$$

et aux  $2n$  contraintes inégalités :

$$\mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u \quad (6.2)$$

qui minimise le critère :

$$c = \frac{1}{2} \|\mathbf{x} - \mathbf{x}^o\|^2 \quad (6.3)$$

où les composantes de  $\mathbf{x}^l$  et  $\mathbf{x}^u$  sont les bornes inférieures et supérieures des  $x_i$  et où  $\mathbf{x}^o$  est une valeur *préférée* de  $\mathbf{x}$ . La matrice  $\mathbf{A}$  est supposée de rang complet, à savoir :

$$\text{rang}(\mathbf{A}) = \dim(\mathbf{b}) = m$$

La méthode que nous proposons est adaptée à des problèmes pour lesquels la solution optimale  $\hat{\mathbf{x}}^s$  du problème sans les contraintes inégalités est souvent solution du problème avec contraintes. Cette solution est calculée directement par la formule (6.7) donnée ci-après. Si elle vérifie les contraintes (6.2) la solution optimale est trouvée. Dans le cas contraire, les contraintes violées sont tour à tour prises en compte. L'algorithme présenté donne d'excellents résultats (par rapport à d'autres) dans le cas où la redondance :

$$r = n - m$$

est faible. Dans ce cas le nombre de combinaisons de contraintes qui sont simultanément saturées est relativement faible, et l'ordre judicieux des tests conduit rapidement à la solution optimale.

### 6.1 Optimum sans contraintes inégalités :

On considère le lagrangien :

$$L^s = \frac{1}{2} (\mathbf{x} - \mathbf{x}^o)^T (\mathbf{x} - \mathbf{x}^o) + \lambda^T (\mathbf{Ax} - \mathbf{b}) \quad (6.4)$$

Il est stationnaire quand :

$$\frac{\partial L^s}{\partial \mathbf{x}} = \mathbf{x}^s - \mathbf{x}^o + \mathbf{A}^T \lambda = \mathbf{0} \quad (6.5)$$

ce qui implique :

$$\mathbf{x}^s = \mathbf{x}^o - \mathbf{A}^T \lambda \quad (6.6)$$

Compte tenu de la contrainte égalité 6.1, on a :

$$\mathbf{b} = \mathbf{A}\mathbf{x}^o - \mathbf{A}\mathbf{A}^T \lambda$$

soit :

$$\lambda = -(\mathbf{A}\mathbf{A}^T)^{-1} (\mathbf{b} - \mathbf{A}\mathbf{x}^o)$$

car  $\mathbf{A}$  est de rang complet. Reportons  $\lambda$  dans 6.6. Il vient :

$$\mathbf{x}^s = \mathbf{x}^o + \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} (\mathbf{b} - \mathbf{A}\mathbf{x}^o) \quad (6.7)$$

Si  $\mathbf{x}^s$  vérifie les inégalités (6.2), c'est la solution au problème posé.

Remarque : Au niveau calcul numérique, on peut éviter le calcul de  $(\mathbf{A}\mathbf{A}^T)^{-1}$ . On calcule  $\lambda$  solution du système symétrique  $(\mathbf{A}\mathbf{A}^T) \lambda = \mathbf{A}\mathbf{x}^o - \mathbf{b}$ , puis on reporte cette valeur dans (6.6). Toutefois si de nombreuses contraintes sont susceptibles d'être saturées dans la solution finale, il y a intérêt à calculer la matrice  $(\mathbf{A}\mathbf{A}^T)^{-1}$  car elle risque d'être utilisée sur plus de  $m$  vecteurs dans la suite de l'algorithme.

#### Interprétation géométrique :

Si  $\mathbf{x}^o = \mathbf{0}$ , le problème consiste à trouver la solution de norme minimale au système  $\mathbf{A}\mathbf{x} = \mathbf{b}$ . Cette solution s'écrit (cf.6.6)  $\mathbf{x}^{\min} = -\mathbf{A}^T \lambda$  qui montre que la solution de norme minimale est une combinaison linéaire des vecteurs lignes  $\mathbf{l}_i$  de la matrice  $\mathbf{A}$  et les  $\lambda_i$  sont (au signe près) les coefficients de cette combinaison.

Lorsque  $\mathbf{x}^o \neq \mathbf{0}$ , la solution s'écrit :

$$\mathbf{x}^s = \mathbf{x}^{\min} + \mathbf{n}, \text{ avec } \mathbf{n} = \left( \mathbf{I} - \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A} \right) \mathbf{x}^o$$

où  $\mathbf{n}$  est la projection orthogonale  $\mathbf{x}^o$  dans le noyau de  $\mathbf{A}$  (cf. paragraphe 2.2.2 du mémo "Algorithmes numériques pour l'algèbre linéaire").

## 6.2 Optimum avec contraintes inégalités

Si  $\mathbf{x}^s$  ne vérifie pas une ou plusieurs des inégalités (6.2) on cherche de nouvelles solutions en saturant une ou plusieurs contraintes parmi celles qui sont violées par  $\mathbf{x}^s$  ou par les solutions ultérieures.

### 6.2.1 Saturation d'une seule contrainte

Supposons par exemple que  $x_k^s > x_k^u$ , c'est-à-dire que  $x_k^o + \mathbf{a}_k^T (\mathbf{A}\mathbf{A}^T)^{-1} (\mathbf{b} - \mathbf{A}\mathbf{x}^o) > x_k^u$  où  $\mathbf{a}_k$  désigne le  $k$ -ième vecteur colonne de  $\mathbf{A}$ .

Cherchons la solution où on impose la contrainte supplémentaire  $x_k = x_k^u$  que l'on écrira :

$$\mathbf{e}_k^T \mathbf{x} = x_k^u \quad (6.8)$$

avec :

$$\mathbf{e}_k^T = ( 0 \quad \dots \quad 0 \quad 1 \quad 0 \quad \dots \quad 0 )$$

le 1 étant en  $k$ -ième position.

Le nouveau lagrangien s'écrit :

$$L = \frac{1}{2} (\mathbf{x} - \mathbf{x}^o)^T (\mathbf{x} - \mathbf{x}^o) + \lambda^T (\mathbf{A}\mathbf{x} - \mathbf{b}) + \mu_k (\mathbf{e}_k^T \mathbf{x} - x_k^u)$$

où  $\mu_k$  est un paramètre de Kuhn et Tucker qui doit être positif.

La stationnarité du gradient s'écrit :

$$\frac{\partial L}{\partial \mathbf{x}} = \mathbf{x} - \mathbf{x}^o + \mathbf{A}^T \lambda + \mu_k \mathbf{e}_k = 0 \quad (6.9)$$

soit :

$$\mathbf{x} = \mathbf{x}^o - \mathbf{A}^T \lambda - \mu_k \mathbf{e}_k \quad (6.10)$$

Les contraintes 6.1 et 6.8 donnent :

$$\mathbf{b} - \mathbf{A}\mathbf{x}^o + \mathbf{A}\mathbf{A}^T \lambda + \mu_k \mathbf{a}_k = 0 \quad (6.11)$$

et

$$x_k^u - x_k^o + \mathbf{a}_k^T \lambda + \mu_k = 0 \quad (6.12)$$

### Première méthode de résolution :

Cette méthode n'est citée qu'à titre de curiosité, car elle s'est avérée moins intéressante que la suivante.

De 6.12 on tire le paramètre de Kuhn et Tucker :

$$\mu_k = x_k^o - x_k^u - \mathbf{a}_k^T \lambda \quad (6.13)$$

qui, reporté dans 6.11 donne :

$$\mathbf{b} - \mathbf{A}\mathbf{x}^o + \mathbf{A}\mathbf{A}^T \lambda + \mathbf{a}_k (x_k^o - x_k^u - \mathbf{a}_k^T \lambda) = \mathbf{0}$$

D'où :

$$\lambda = - [\mathbf{A}\mathbf{A}^T - \mathbf{a}_k \mathbf{a}_k^T]^{-1} [\mathbf{b} - \mathbf{A}\mathbf{x}^o + \mathbf{a}_k (x_k^o - x_k^u)] \quad (6.14)$$

ce qui suppose que

$$\mathbf{A}\mathbf{A}^T - \mathbf{a}_k \mathbf{a}_k^T = \sum_{i \neq k} \mathbf{a}_i \mathbf{a}_i^T$$

est de rang complet, c'est-à-dire que la matrice  $\mathbf{A}$  sans la colonne  $\mathbf{a}_k$  soit de rang complet.

En reportant la valeur de  $\lambda$  dans l'expression de  $\mu_k$  on obtient :

$$\mu_k = x_k^o - x_k^u + \mathbf{a}_k^T [\mathbf{A}\mathbf{A}^T - \mathbf{a}_k \mathbf{a}_k^T]^{-1} [\mathbf{b} - \mathbf{A}\mathbf{x}^o + \mathbf{a}_k (x_k^o - x_k^u)] \quad (6.15)$$

En reportant 6.14 et 6.15 dans 6.10 on obtient :

$$\begin{aligned} \mathbf{x} &= \mathbf{x}^o + \mathbf{A}^T (\mathbf{A}\mathbf{A}^T - \mathbf{a}_k \mathbf{a}_k^T)^{-1} [\mathbf{b} - \mathbf{A}\mathbf{x}^o + \mathbf{a}_k (x_k^o - x_k^u)] \text{ pour } i \neq k \\ x_k &= x_k^u \text{ pour } i = k \end{aligned}$$

### Deuxième méthode de résolution :

Notons  $\mathbf{x}^s$  et  $\lambda^s$  les solutions du problème sans contrainte qui sont telles que :

$$\begin{aligned} \mathbf{x}^s - \mathbf{x}^o + \mathbf{A}^T \lambda^s &= \mathbf{0} \\ \mathbf{A}\mathbf{x}^s &= \mathbf{b} \\ x_k^s &> x_k^u \end{aligned}$$

En les prenant en compte, les équations à résoudre s'écrivent :

$$\begin{aligned} (\mathbf{x} - \mathbf{x}^s) + \mathbf{A}^T (\lambda - \lambda^s) + \mu_k \mathbf{e}_k &= \mathbf{0} \\ \mathbf{A}(\mathbf{x} - \mathbf{x}^s) &= \mathbf{0} \\ \mathbf{e}_k^T (\mathbf{x} - \mathbf{x}^s) &= x_k^u - x_k^s \end{aligned}$$

En multipliant la première relation par  $\mathbf{A}$  puis par  $\mathbf{e}_k^T$  il vient :

$$\begin{aligned}\mathbf{A}\mathbf{A}^T(\lambda - \lambda^s) + \mu_k \mathbf{a}_k &= \mathbf{0} \\ (x_k^u - x_k^s) + \mathbf{a}_k^T(\lambda - \lambda^s) + \mu_k &= 0\end{aligned}$$

De la première on tire :

$$\lambda = \lambda^s - \mu_k (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{a}_k \quad (6.16)$$

qui reporté dans la deuxième donne :

$$(x_k^u - x_k^s) - \mathbf{a}_k^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{a}_k \mu_k + \mu_k = 0$$

soit :

$$\mu_k = \delta_k (x_k^s - x_k^u) \text{ avec } \delta_k = \frac{1}{1 - \mathbf{a}_k^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{a}_k} \quad (6.17)$$

Cette valeur reportée dans 6.16 fournit  $\lambda$  qui, reporté dans 6.10 donne :

$$\mathbf{x} = \mathbf{x}^s + \delta_k (x_k^s - x_k^u) \left[ \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{a}_k - \mathbf{e}_k \right] \quad (6.18)$$

On remarquera que :

$$x_k = x_k^s + \delta_k (x_k^s - x_k^u) \left[ \mathbf{a}_k^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{a}_k - 1 \right] = x_k^u$$

Cette deuxième méthode à l'avantage de se prêter à diverses interprétations géométriques.

Montrons en premier lieu que  $\mu_k \geq 0$ .

Le système linéaire redondant  $\mathbf{w} = \mathbf{A}\mathbf{v} = \sum_i \mathbf{a}_i v_i$ , a pour solution qui minimise  $\|\mathbf{v}\|$  :

$$\mathbf{v} = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{w}$$

Dans le cas où  $\mathbf{w}_k = \mathbf{a}_k$ , on a une solution évidente  $\mathbf{v}_k = \mathbf{e}_k$ , telle que  $\|\mathbf{v}_k\| = 1$ . Si  $\hat{\mathbf{v}}_k$  est de norme minimale, toutes ses composantes sont inférieures ou égales à  $\|\mathbf{v}_k\|$ . En particulier :

$$\hat{v}_k^k = \mathbf{a}_k^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{a}_k \leq 1 \rightarrow \delta_k \geq 0$$

Comme  $x_k^s > x_k^u$ , il en résulte que  $\mu_k \geq 0$ .

Remarque :  $\hat{\mathbf{v}}_k$  est le  $k$ -ième vecteur colonne du projecteur orthogonal  $\mathbf{P} = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}$  sur les lignes de  $\mathbf{A}$ .

Les relations 6.5 et 6.9 montrent que  $\mu_k \mathbf{e}_k = -(\mathbf{x} - \mathbf{x}^o + \mathbf{A}^T \lambda) = -\frac{\partial L^s}{\partial \mathbf{x}}$ , soit :

$$\mu_k = -\frac{\partial L^s}{\partial x_k} \rightarrow \delta L^s = -\mu \delta x_k$$

$x_k^s$  est saturé à  $x_k^u$ . Si on le dé-sature, on ne peut que le diminuer,  $\delta x_k$  est négatif et  $\delta L^s$  est positif, le critère augmente comme on pouvait s'y attendre puisque l'optimum sans contrainte est au delà de  $x_k^u$ .

Remarque : Si les  $\mathbf{a}_i$  sont des directions de l'espace et si les  $\mathbf{v}_i$  sont les forces élémentaires exercées sur ces directions, alors  $\mathbf{w}$  est la force résultante. Si on considère une force résultante alignée sur  $\mathbf{a}_k$ , alors la force élémentaire sur la direction  $\mathbf{a}_k$  est égale à  $\hat{v}_k^k$  (pour la solution  $\hat{\mathbf{v}}_k$  de norme minimale). Le terme  $1 - \hat{v}_k^k$  est la différence entre la solution où seul  $\mathbf{a}_k$  fournit l'effort et la solution de norme minimale, c'est-à-dire le gain d'effort fait sur cette direction (il y a une perte sur toutes les autres). Quand pour



réaliser l'effort résultant  $\mathbf{b}$ , l'effort sur  $\mathbf{a}_k$  dépasse l'effort maximal  $x_k^u$  de  $\Delta x_k = x_k^s - x_k^u$ , le paramètre de Kuhn et Tucker, utilisé pour saturer cette composante et uniquement celle-ci, vaut  $\mu_k = \Delta x_k / (1 - \hat{v}_k^k)$ .

La solution 6.18 s'écrit :

$$\mathbf{x} = \mathbf{x}^s + \frac{(x_k^s - x_k^u)}{1 - \hat{v}_k^k} [\hat{\mathbf{v}}_k - \mathbf{e}_k] \quad (6.19)$$

d'où :

$$\|\mathbf{x} - \mathbf{x}^s\| = \frac{|x_k^s - x_k^u|}{1 - \hat{v}_k^k} \|\hat{\mathbf{v}}_k - \mathbf{e}_k\| \quad (6.20)$$

### Cas de la contrainte inférieure

Supposons que la contrainte  $x_k^s \geq x_k^l$  soit violée (autrement dit  $x_k^s < x_k^l$ ). Dans ce cas on sature  $x_k$  à  $x_k^l$ . On remplace dans les calculs précédents  $x_k^u$  par  $x_k^l$ . Tous les calculs sont identiques. Seul diffère le signe du paramètre de Kuhn et Tucker qui sera négatif :

$$\mu_k \leq 0$$

## 6.2.2 Saturation de plusieurs contraintes

Supposons que l'on sature simultanément les contraintes  $k_1, \dots, k_q$ .

Pour  $q > r = n - m$  le système est trop contraint. Dans ce cas la stratégie de résolution dépend de l'importance relative entre contraintes. On peut prendre en contraintes égalités les  $r$  contraintes critiques pour la sécurité, et on minimisera la somme des carrés des distances pondérées aux autres.

Pour  $q = r$ ,  $\mathbf{x}$  est exactement imposé par les  $m + r = n$  contraintes.

Supposons  $q < r$  et appliquons la deuxième méthode de résolution.

Le lagrangien s'écrit :

$$L = \frac{1}{2} (\mathbf{x} - \mathbf{x}^o)^T (\mathbf{x} - \mathbf{x}^o) + \lambda^T (\mathbf{A}\mathbf{x} - \mathbf{b}) + \boldsymbol{\mu}^T (\mathbf{E}^T \mathbf{x} - \mathbf{s})$$

avec :

$$\boldsymbol{\mu}^T = ( \mu_{k_1} \quad \dots \quad \mu_{k_q} ) , \quad \mathbf{E} = ( \mathbf{e}_{k_1} \quad \dots \quad \mathbf{e}_{k_q} ) \quad \text{et} \quad \mathbf{s} = \begin{pmatrix} x_{k_1}^{\text{lim}} \\ \vdots \\ x_{k_q}^{\text{lim}} \end{pmatrix}$$

Pour  $x_{k_i}^{\text{lim}} = x_{k_i}^l$ , il faudra vérifier que  $\mu_{k_i}$  est négatif et pour  $x_{k_i}^{\text{lim}} = x_{k_i}^u$ , il faudra vérifier que  $\mu_{k_i}$  est positif.

La stationnarité du gradient de  $L$  s'écrit :

$$\frac{\partial L}{\partial \mathbf{x}} = \mathbf{x} - \mathbf{x}^o + \mathbf{A}^T \boldsymbol{\lambda} + \mathbf{E} \boldsymbol{\mu} = \mathbf{0}$$

d'où :

$$\mathbf{x} - \mathbf{x}^s + \mathbf{A}^T (\boldsymbol{\lambda} - \boldsymbol{\lambda}^s) + \mathbf{E} \boldsymbol{\mu} = \mathbf{0}$$

qui multiplié par  $\mathbf{A}$  puis par  $\mathbf{E}^T$  donne :

$$\begin{aligned} \mathbf{A} \mathbf{A}^T (\boldsymbol{\lambda} - \boldsymbol{\lambda}^s) + \mathbf{A}_s \boldsymbol{\mu} &= \mathbf{0} \\ -\Delta \mathbf{s} + \mathbf{A}_s^T (\boldsymbol{\lambda} - \boldsymbol{\lambda}^s) + \boldsymbol{\mu} &= \mathbf{0} \end{aligned}$$

où on a posé :

$$\begin{aligned} \Delta \mathbf{s} &= (\mathbf{E}^T \mathbf{x}^s - \mathbf{s}) \\ \mathbf{A}_s &= \mathbf{A} \mathbf{E} = ( \mathbf{a}_{k_1} \quad \dots \quad \mathbf{a}_{k_q} ) \end{aligned}$$

On en tire :

$$\begin{aligned} (\lambda - \lambda^s) &= -(\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}_s \mu \\ -\Delta \mathbf{s} + \left[ \mathbf{I} - \mathbf{A}_s^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}_s \right] \mu &= \mathbf{0} \end{aligned}$$

D'où :

$$\mu = \left[ \mathbf{I} - \mathbf{A}_s^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}_s \right]^{-1} \Delta \mathbf{s}$$

et :

$$\mathbf{x} = \mathbf{x}^s + \left[ \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}_s - \mathbf{E} \right] \left[ \mathbf{I} - \mathbf{A}_s^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}_s \right]^{-1} \Delta \mathbf{s}$$

Il faut vérifier que les composantes de  $\mu$  ont le bon signe.

Remarque pour les calculs numériques :

1. Le vecteur  $\mu = \left[ \mathbf{I} - \mathbf{A}_s^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}_s \right]^{-1} \Delta \mathbf{s}$  est calculé en résolvant le système symétrique d'ordre  $q$  :  $\left[ \mathbf{I} - \mathbf{A}_s^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}_s \right] \mu = \Delta \mathbf{s}$
2. Les colonnes des différentes matrices  $(\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}_s$  (pour différentes matrices  $\mathbf{A}_s$ ) sont un sous-ensemble des colonnes de  $(\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}$ . Chaque fois que l'une d'entre elle est calculée, elle est mémorisée pour ne pas être calculée une deuxième fois.
3. En notant  $\mathbf{V}^{k_i}$  les colonnes de  $\mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}_s$ . La solution s'écrit  $\mathbf{x} = \mathbf{x}^s + \sum_{i=1}^q (\mathbf{V}^{k_i} - \mathbf{e}_{k_i}) \mu_{k_i}$

### 6.2.3 Stratégie de saturation

Elle est inspirée de la méthode de Theil et van de Panne<sup>1</sup>. Si la solution sans contrainte  $\mathbf{x}^s$  en viole plusieurs, et s'il existe une solution optimale avec une seule contrainte saturée, alors la relation 6.20 montre que cette contrainte est forcément celle qui maximise  $\frac{|x_k^s - x_k^{\text{lim}}|}{1 - \hat{v}_k^k} \|\hat{\mathbf{v}}_k - \mathbf{e}_k\|$ . Si les  $|\Delta x_k| = |x_k^s - x_k^{\text{lim}}|$  sont aisés à calculer ces écarts sont pondérés par les facteurs  $\frac{\|\hat{\mathbf{v}}_k - \mathbf{e}_k\|}{1 - \hat{v}_k^k}$  (quelconques a priori) qui, une fois calculés, donnent immédiatement les solutions contraintes. Il semble qu'il n'y ait pas de stratégie qui permette de trouver directement quelle est la contrainte dont la saturation satisfasse toutes les autres, quand elle existe.

On sature donc les contraintes, une à une, dans un ordre quelconque (éventuellement dans l'ordre du classement des  $|\Delta x_k|$  du plus grand au plus petit).

- Si une solution avec une seule contrainte saturée, satisfait les autres contraintes, elle est optimale, il est inutile d'en chercher d'autres.
- Sinon, il faut saturer deux contraintes. On considère tous les couples possibles. On en choisit un.

Si les paramètres de Kuhn et Tucker des deux contraintes saturées ont le bon signe :

- Si les autres contraintes sont satisfaites, la solution est optimale, il est inutile d'en chercher d'autres.
- Sinon, on mémorise les triplets possibles.

On examine le couple suivant, jusqu'à trouver un couple bon.

- Si aucune solution saturant deux contraintes n'est bonne, on recommence la procédure avec les triplets et ainsi de suite, jusqu'à ce que le nombre de variables que l'on peut saturer atteigne  $r$ .

1. Theil, H. et Van de Panne, "Quadratic Programming as an Extension of Conventional Quadratic Maximization." Management Science, Vol. 7, 1960.

- Si aucune solution n'est bonne quand on sature simultanément  $r$  contraintes, le problème n'a pas de solution exacte. On peut alors chercher une solution approchée, par exemple au sens des moindres carrés.



## Chapitre 7

# PNL sans contraintes

Considérons le problème de programmation non-linéaire (PNL) sans contraintes suivant :

$$\min f(\mathbf{x}) \in \mathbb{R}, \quad \mathbf{x} \in \mathbb{R}^n$$

où  $f(\mathbf{x})$  possède une approximation au 2ème ordre donnée par :

$$f(\mathbf{x} + \mathbf{d}) = f(\mathbf{x}) + \mathbf{f}_x^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \mathbf{F}_{xx} \mathbf{d}$$

avec :

$$\mathbf{f}_x = \nabla f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \text{ gradient de } f \text{ relativement à } \mathbf{x}$$

$$\mathbf{F}_{xx} = \nabla^2 f(\mathbf{x}) = \begin{cases} \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}^2} \text{ Hessienne de } f \text{ relativement à } \mathbf{x} \\ \nabla \mathbf{f}_x = \frac{\partial \mathbf{f}_x}{\partial \mathbf{x}} \text{ Jacobienne de } \mathbf{f}_x \text{ relativement à } \mathbf{x} \end{cases}$$

L'approximation au premier ordre du gradient  $\mathbf{f}_x$  est donnée par :

$$\mathbf{f}_x(\mathbf{x} + \mathbf{d}) = \mathbf{f}_x(\mathbf{x}) + \mathbf{F}_{xx} \mathbf{d}$$

Les conditions d'optimalité du premier et deuxième ordre nous indiquent que  $\hat{\mathbf{x}}$  est un minimum local si :

1.  $\nabla f(\hat{\mathbf{x}}) = 0$
2.  $\nabla^2 f(\hat{\mathbf{x}})$  est définie positive.

### 7.1 Cas monovariante

A développer : Exploration systématique, recherche par dichotomie, découpage des longueurs des segments successifs de recherche selon la suite de fibonacci, selon le rapport nombre d'or,....

### 7.2 Cas multivariante

#### 7.2.1 Méthodes d'ordre zéro

A développer :

- Méthode de Gauss : optimisation monovariante, tour à tour sur chaque variable,
- Méthode du simplexe de Nelder-Mead : test d'un lot de  $n + 1$  points, suppression du plus mauvais et remplacement par un nouveau point situé du côté opposé du point supprimé par rapport à l'hyperplan défini par les points conservés,...

### 7.2.2 Méthodes d'ordre 1 : Algorithmes de descente

La recherche de ce minimum de fait à l'aide d'algorithmes de descentes, à savoir par la recherche d'incrément successifs de descente  $\theta \mathbf{d}$ , avec  $\theta > 0$  et  $\|\mathbf{d}\| = 1$ , tels que :  $f(\mathbf{x} + \theta \mathbf{d}) < f(\mathbf{x})$  ce qui implique que  $\theta \mathbf{f}_x^T \mathbf{d} < 0$ .

$$f(\mathbf{x} + \theta \mathbf{d}) < f(\mathbf{x}), \text{ avec } \theta > 0, \|\mathbf{d}\| = 1 \rightarrow \theta \mathbf{f}_x^T \mathbf{d} < 0$$

#### Conditions de Wolfe sur la direction $\mathbf{d}$

Une direction est considérée suffisamment descendante s'il existe deux constantes positives  $\gamma_0$  et  $\gamma_1$  indépendantes de  $x$  telle que  $\mathbf{d}$  satisfasse les inégalités suivantes :

$$\begin{aligned} \mathbf{d}^T \mathbf{f}_x(\mathbf{x}) &\leq -\gamma_0 \mathbf{f}_x^T \mathbf{f}_x \\ \mathbf{d}^T \mathbf{d} &\leq \gamma_1 \mathbf{f}_x^T \mathbf{f}_x \end{aligned}$$

La première indique que  $\mathbf{d}$  fait un angle assez aigu avec  $-\mathbf{f}_x^T$  et la deuxième que  $\mathbf{d}$  n'est pas trop long.

L'opposé du gradient  $\mathbf{d} = -\mathbf{f}_x$  (utilisé dans la méthode du gradient) satisfait les deux conditions pour  $\gamma_0 = 1$  et  $\gamma_1 = 1$  (par exemple).

#### Conditions sur le pas

Un pas  $\theta$  est satisfaisant si :

$$\begin{aligned} f(\mathbf{x} + \theta \mathbf{d}) - f(\mathbf{x}) &\leq \tau_0 \theta \mathbf{d}^T \mathbf{f}_x(\mathbf{x}) \text{ avec } \tau_0 \in (0, \frac{1}{2}) \text{ (critère d'Armijo)} \\ \mathbf{d}^T \mathbf{f}_x(\mathbf{x} + \theta \mathbf{d}) &\geq \tau_1 \mathbf{d}^T \mathbf{f}_x(\mathbf{x}) \text{ avec } \tau_1 \in (0, 1) \text{ (condition de Wolfe)} \end{aligned}$$

Le critère d'Armijo assure que le pas  $\theta$  fait suffisamment diminuer  $f$  (on se limite généralement à des valeurs faibles de  $10^{-4}$  à  $10^{-2}$ ) et la condition de Wolfe que la courbure correspond à un minimum (on prend généralement  $\tau_1 \simeq 0.9$  pour les méthodes inspirées de la méthode de Newton et  $\tau_1 \simeq 0.1$  pour le gradient conjugué). Cette deuxième condition est souvent remplacée par la condition "forte" de Wolfe :

$$|\mathbf{d}^T \mathbf{f}_x(\mathbf{x} + \theta \mathbf{d})| \leq \tau_1 |\mathbf{d}^T \mathbf{f}_x(\mathbf{x})| \text{ avec } \tau_1 \in (0, 1)$$

#### Règle d'Armijo :

A chaque itération on choisit une direction  $\mathbf{d}$  suffisamment descendante et pour  $\theta$  la première valeur dans la suite  $\{1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{2^n}, \dots\}$  qui satisfait le critère d'Armijo.

#### Méthode du gradient

On choisit la direction  $\mathbf{d}$  selon la pente la plus forte, à savoir l'opposé du gradient :

$$\mathbf{d} = -\mathbf{f}_x$$

puis on cherche  $\theta$  qui minimise  $f(\mathbf{x} + \theta \mathbf{d})$  ou en appliquant la règle d'Armijo. On vérifiera que dans le cas quadratique ( $f(\mathbf{x}) = c + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}$ ), la valeur théorique qui annule  $\mathbf{f}_x(\mathbf{x} + \theta \mathbf{d})$  est :

$$\theta = \frac{\|\mathbf{f}_x\|^2}{\mathbf{f}_x^T \mathbf{F}_{xx} \mathbf{f}_x}$$

### Méthode du gradient conjugué

La méthode standard du gradient est relativement lente car les directions  $\mathbf{d}_k$  de descentes successives sont orthogonales (puisque  $\theta_k$  est tel que  $\mathbf{d}_k$  est tangent en  $\mathbf{x}_{k+1}$  à la surface de niveau et que  $\mathbf{d}_{k+1}$  est orthogonal à cette surface de niveau). Les directions successives de la méthode dite du gradient conjugué sont a priori mieux disposées.

On initialise par  $\mathbf{d}_0 = -\mathbf{f}_x(\mathbf{x}_0)$ , mais les  $\mathbf{d}_k$  suivants sont pris égaux à :

$$\mathbf{d}_k = -\mathbf{f}_x(\mathbf{x}_k) + \frac{\|\mathbf{f}_x(\mathbf{x}_k)\|^2}{\|\mathbf{f}_x(\mathbf{x}_{k-1})\|^2} \mathbf{d}_{k-1}$$

Dans le cas quadratique ( $f(\mathbf{x}) = c + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}$ ), on prend  $\theta = \frac{\|\mathbf{f}_x\|^2}{\mathbf{d}^T \mathbf{F}_{xx} \mathbf{d}}$  et il en résulte que les directions  $\mathbf{d}_k$  successives sont conjuguées (telles que  $\mathbf{d}_k^T \mathbf{A} \mathbf{d}_{k+1} = 0$ , c'est-à-dire orthogonales pour le produit scalaire  $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{A} \mathbf{v}$ ). Dans ce cas la convergence se fait en  $n$  coups ( $n$  est la dimension de  $\mathbf{x}$ ).

Dans le cas non-linéaire **Polak-Ribière** proposent la variante :

$$\mathbf{d}_k = -\mathbf{f}_x(\mathbf{x}_k) + \frac{\mathbf{f}_x^T(\mathbf{x}_k) (\mathbf{f}_x(\mathbf{x}_k) - \mathbf{f}_x(\mathbf{x}_{k-1}))}{\|\mathbf{f}_x(\mathbf{x}_{k-1})\|^2} \mathbf{d}_{k-1}$$

### 7.2.3 Méthode d'ordre 2

#### Méthode de Newton

Dans l'approximation quadratique, le gradient est annulé en seul coup au nouveau point :

$$\mathbf{f}_x(\mathbf{x} + \mathbf{d}) = \mathbf{f}_x(\mathbf{x}) + \mathbf{F}_{xx} \mathbf{d} = 0 \Rightarrow \mathbf{d} = -\mathbf{F}_{xx}^{-1} \mathbf{f}_x$$

#### Variantes par approximation du hessien ou de son inverse

On approxime  $\mathbf{F}_{xx}(\mathbf{x}_k)$  par une matrice  $\mathbf{B}_k$  ou son inverse par une matrice  $\mathbf{H}_k$  calculées par une formule récurrente ou son inverse

On note :

$$\begin{aligned} \mathbf{s}_k &= \mathbf{x}_{k+1} - \mathbf{x}_k \\ \mathbf{y}_k &= \mathbf{f}_x(\mathbf{x}_{k+1}) - \mathbf{f}_x(\mathbf{x}_k) \end{aligned}$$

On initialise  $\mathbf{B}_0$  définie positive ( $\mathbf{I}_{n \times n}$  par exemple) et  $\mathbf{f}_x(\mathbf{x}_0)$ .

1.  $\mathbf{d}_k = -\mathbf{B}_k^{-1} \mathbf{f}_x(\mathbf{x}_k)$  ou  $\mathbf{d}_k = -\mathbf{H}_k \mathbf{f}_x(\mathbf{x}_k)$  : direction ligne de recherche.
2. Recherche  $\alpha_k$  tel que  $f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) = \min_{\alpha > 0} f(\mathbf{x}_k + \alpha \mathbf{d}_k)$  ;  $\mathbf{s}_k = \alpha_k \mathbf{d}_k$  ;  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$  ;
3. Calcul  $\mathbf{f}_x(\mathbf{x}_{k+1})$ , fin si suffisamment petit.
4.  $\mathbf{y}_k = \mathbf{f}_x(\mathbf{x}_{k+1}) - \mathbf{f}_x(\mathbf{x}_k)$

5.  $\mathbf{B}_{k+1} = \mathcal{F}(\mathbf{B}_k, \mathbf{s}_k, \mathbf{y}_k)$  ou  $\mathbf{H}_{k+1} = \mathcal{G}(\mathbf{H}_k, \mathbf{s}_k, \mathbf{y}_k)$
6.  $k \leftarrow k + 1$ , retour étape 1.

Il existe de nombreuses équations récurrentes de mise à jour. Citons :

**Méthode DFP : Davidon (1959), Fletcher & Powell (1963)**

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} - \frac{\mathbf{H}_k \mathbf{y}_k \mathbf{y}_k^T \mathbf{H}_k}{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k}$$

**Méthodes BFGS (Broyden, Fletcher, Goldfarb & Shanno)**

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k}$$

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \left(1 + \frac{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k}{\mathbf{s}_k^T \mathbf{y}_k}\right) \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} - \left(\frac{\mathbf{s}_k \mathbf{y}_k^T \mathbf{H}_k + \mathbf{H}_k \mathbf{y}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}\right)$$

### 7.2.4 Méthodes géométriques

A faire : méthode de Partan (gradient conjugué)



## Chapitre 8

# PNL avec contraintes (A Faire)

A développer :

### 8.1 Méthode du gradient projeté

A développer :

1. Départ d'un point  $\mathbf{x}$  satisfaisant l'ensemble  $\mathcal{C}$  des contraintes (problème d'initialisation) et en saturant éventuellement quelques unes.
2. Recherche de l'ensemble  $\mathcal{K}$  des contraintes saturées qui sont en nombre  $k$ .
3. Recherche parmi les  $2^k$  sous-ensembles  $\mathcal{K}^s$  de  $\mathcal{K}$  (incluant l'ensemble vide et  $\mathcal{K}$  lui-même) du sous ensemble  $\mathcal{K}_{opt}^s$  tel que le déplacement selon la projection  $\delta$  du gradient du critère  $\mathbf{g}$  sur le sous-espace intersection des espaces tangents aux contraintes de  $\mathcal{K}_{opt}^s$  :

ne viole pas (au premier ordre, au niveau des plans tangents) les contraintes du sous-ensemble complémentaire  $\mathcal{K}^{c1} = \mathcal{K} - \mathcal{K}_{opt}^s$ ,

et produise la plus grande variation du critère.

4. Si la dimension de  $\mathcal{K}_{opt}^s \geq \dim(\mathbf{x})$  la recherche est terminée car le sous-espace intersection est un point et la projection  $\delta$  n'existe plus. On a atteint un minimum local.
5. Sinon, déplacement depuis  $\mathbf{x}$  selon la direction  $\delta$  d'une distance limitée de telle manière que le sous-ensemble  $\mathcal{K}_{opt}^s$  reste saturé et qu'aucune contrainte du sous-ensemble  $\mathcal{K}^{c2} = \mathcal{C} - \mathcal{K}_{opt}^s$  ne soit violée. On cherche à saturer une contrainte du sous-ensemble  $\mathcal{K}^{c3} = \mathcal{C} - \mathcal{K}$ .
6. Retour en 2.

La phase 5 est la phase critique du gradient projeté. Il faut régler le pas de manière à ne pas violer les contraintes du sous-ensemble  $\mathcal{K}^{c2}$ , et s'approcher le plus des contraintes du sous-ensemble  $\mathcal{K}^{c3}$  (celles qui n'étaient pas saturées). En fait, la non-linéarité des contraintes de  $\mathcal{K}_{opt}^s$  fait que le déplacement va les désaturer (en plus ou en moins). Le pas doit en tenir pour que cette désaturation reste acceptable. A l'issue du pas, on sature à nouveau toutes les contraintes de  $\mathcal{K}_{opt}^s$  et éventuellement celles de  $\mathcal{K}^{c1}$  qui ont été violées à l'aide de l'algorithme de Newton-Ralphson par exemple. Le nouveau point  $\mathbf{x}^+$  projeté peut éventuellement violer une contrainte de  $\mathcal{K}^{c3}$ , ce qui signifie que le pas a été trop grand. On recommence alors la phase 5 avec pour direction  $\delta = \mathbf{x}^+ - \mathbf{x}$ . Cette phase peut donc demander plusieurs itérations.

Si tout se passe bien, à chaque nouvelle grande itération (passant par l'étape 2), le nombre  $k$  de contraintes saturées augmente d'une unité. La sortie de l'algorithme se fait à l'étape 4 quand le point  $\mathbf{x}$

est à l'intersection d'un ensemble d'au moins  $dim(\mathbf{x})$  contraintes avec un gradient dirigé vers l'extérieur du domaine admissible.

## 8.2 Fonctions de pénalisation

A développer : On remplace les contraintes par l'ajout de fonctions de pénalisation dans le critère, fonctions qui le dégradent rapidement quand le point courant s'approche d'une frontière ou qu'il la franchit et qui est sans effet quand il est dans une zone admissible, assez loin de la frontière.