



HAL
open science

Méthodes de calcul des fonctions "spline" dans un convexe

Madeleine Morin

► **To cite this version:**

Madeleine Morin. Méthodes de calcul des fonctions "spline" dans un convexe. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1969. Français. NNT : . tel-00281640

HAL Id: tel-00281640

<https://theses.hal.science/tel-00281640>

Submitted on 23 May 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° D'ORDRE

Tu 2190

THESE
présentée à
LA FACULTE DES SCIENCES DE L'UNIVERSITE DE GRENOBLE
pour obtenir
LE GRADE DE DOCTEUR DE TROISIEME CYCLE
"Mathématiques Appliquées"

par

Madeleine MORIN

licenciée-es-sciences

METHODES DE CALCUL DES
FONCTIONS "SPLINE" DANS UN CONVEXE.

These soutenue le 17 Septembre 1969, devant la Commission d'examen :

Monsieur J. KUNTZMANN :	Président
Messieurs J.R. BARRA N. GASTINEL P.J. LAURENT }	Examineurs

L I S T E D E S P R O F E S S E U R S

Doyen honoraire : Monsieur M. MORET
 Doyen : Monsieur E. BONNIER

PROFESSEURS TITULAIRES

MM.	NEEL Louis	Physique Expérimentale
	KRAVTCHENKO Julien	Mécanique Rationnelle
	CHABAUTY Claude	Calcul différentiel et intégral
	BENOIT Jean	Radioélectricité
	CHENE Marcel	Chimie Papetière
	FELICI Noël	Electrostatique
	KUNTZMANN Jean	Mathématiques Appliquées
	BARBIER Reynold	Géologie Appliquée
	SANTON Lucien	Mécanique des Fluides
	OZENDA Paul	Botanique
	FALLOT Maurice	Physique Industrielle
	KOSZUL Jean-Louis	Mathématiques
	GALVANI Octave	Mathématiques
	MOUSSA André	Chimie Nucléaire
	TRAYNARD Philippe	Chimie Générale
	SOUTIF Michel	Physique Générale
	CRAYA Antoine	Hydrodynamique
	REULOS René	Théorie des Champs
	BESSON Jean	Chimie Minérale
	AYANT Yves	Physique Approfondie
	GALLISSOT François	Mathématiques
Melle.	LUTZ Elisabeth	Mathématiques
MM.	BLAMBERT Maurice	Mathématiques
	BOUCHEZ Robert	Physique Nucléaire
	LLIBOUTRY Louis	Géophysique
	MICHEL Robert	Minéralogie et pétrographie
	BONNIER Etienne	Electrochimie et Electrometallurgie
	DESSAUX Georges	Physiologie animale
	PILLET Emile	Physique Industrielle-Electrotechnique
	YOCCOZ Jean	Physique Nucléaire théorique
	DEBELMAS Jacques	Géologie Générale
	GERBER Robert	Mathématiques
	PAUTHENET René	Electrotechnique
	MALGRANGE Bernard	Mathématiques Pures
	VAUQUOIS Bernard	Calcul Electronique
	BARJON Robert	Physique Nucléaire

MM.	BARBIER Jean-Claude	Physique
	SILBER Robert	Mécanique des Fluides
	BUYLE-BODIN Maurice	Electronique
	DREYFUS Bernard	Thermodynamique
	KLEIN Joseph	Mathématiques
	VAILLANT François	Zoologie et Hydrobiologie
	ARNAUD Paul	Chimie
	SENGEL Philippe	Zoologie
	BARNOUD Fernand	Biosynthèse de la Cellulose
	BRISSONNEAU Pierre	Physique
	GAGNAIRE Didier	Chimie Physique
Mme.	KOFLER Lucie	Botanique
MM.	DEGRANGE Charles	Zoologie
	PEBAY-PEROULA Jean-Claude	Physique
	RASSAT André	Chimie Systématique
	DUCROS Pierre	Cristallographie Physique
	DODU Jacques	Mécanique Appliquée I. U. T.
	ANGLES D'AURIAC Paul	Mécanique des Fluides
	LACAZE Albert	Thermodynamique
	GASTINEL Noël	Analyse numérique
	GIRAUD Pierre	Géologie
	PERRET René	Servo-mécanisme
	PAYAN Jean-Jacques	Mathématiques Pures

PROFESSEURS SANS CHAIRE

MM.	GIDON Paul	Géologie
Mme.	BARBIER Marie-Jeanne	Electrochimie
Mme.	SOUTIF Jeanne	Physique
	COHEN Joseph	Electrotechnique
	DEPASSEL R.	Mécanique des Fluides
	GLENAT René	Chimie
	BARRA Jean	Mathématiques Appliquées
	COUMES André	Electronique
	PERRIAUX Jacques	Géologie et Minéralogie
	ROBERT André	Chimie Papetière
	BIARREZ Jean	Mécanique Physique
	BONNET Georges	Electronique
	CAUQUIS Georges	Chimie Générale
	BONNETAIN Lucien	Chimie Minérale
	DEPOMIER Pierre	Physique Nucléaire-Génie Atomique
	HACQUES Gérard	Calcul numérique
	POLOUJADOFF Michel	Electrotechnique
Mme.	KAHANE Josette	Physique
Mme.	BONNIER Jane	Chimie
MM.	VALENTIN Jacques	Physique
	REBECQ Jacques	Biologie
	DEPORTES Charles	Chimie
	SARROT-REYNAULD Jean	Géologie
	BERTRANDIAS Jean-Paul	Mathématiques Appliquées
	AUBERT Guy	Physique

PROFESSEURS ASSOCIES

MM. RODRIGUES Alexandre Mathématiques Pures
MORITA Susumu Physique Nucléaire
RADHAKRISHNA Thermodynamique

MAITRES DE CONFERENCES

MM. LANCIA Roland Physique Atomique
Mme. BOUCHE Liane Mathématiques
MM. KAHANE André Physique Générale
DOLIQUE Jean Michel Electronique
BRIERE Georges Physique
DESRE Georges Chimie
LAJZEHOWICZ Joseph Physique
LAURENT Pierre Mathématiques Appliquées
Mme. BERTRANDIAS Françoise Mathématiques Pures
MM. LONGEQUEUE Jean-Pierre Physique
SOHM Jean-Claude Electrochimie
ZADWORNÝ François Electronique
DURAND Francis Chimie Physique
CARLIER Georges Biologie végétale
PFISTER Jean-Claude Physique
CHIBON Pierre Biologie animale
IDELMAN Simon Physiologie animale
BLOCH Daniel Electrotechnique I. P.
MARTIN-BOUYER Michel Chimie (C. S. U. Chambéry)
SIBILLE Robert Construction mécanique (I. U. T.)
BRUGEL Lucien Energétique I. U. T.
BOUVARD Maurice Hydrologie
RICHARD Lucien Botanique
PELMONT Jean Physiologie animale
BOUSSARD Jean-Claude Mathématiques Appliquées (I. P. G.)
MOREAU René Hydraulique I. P. G.
ARMAND Yves Chimie I. U. T.
BOLLIET Louis Informatique I. U. T.
KUHN Gérard Energétique I. U. T.
PEFFEN René Chimie I. U. T.
GERMAIN Jean-Pierre Mécanique
JOLY Jean-René Mathématiques Pures
Melle. PIERY Yvette Biologie animale
BERNARD Alain Mathématiques Pures
MOHSEN Tahsin Biologie (C. S. U. Chambéry)
CONTE René Mesures Physiques I. U. T.
LE JUNTER Noël Génie Electrique Electronique I. U. T.
LE ROY Philippe Génie Mécanique I. U. T.
ROMIER Guy Techniques Statistiques quantitatives
I. U. T.
VIALON Pierre Géologie
BENZAKEN Claude Mathématiques Appliquées
MAYNARD Roger Physique

MM.	DUSSAUD René	Mathématiques (C. S. U. Chambéry)
	BELORIZKY Elie	Physique (C. S. U. Chambéry)
Mme.	LAJZEROWICZ Jeannine	Physique (C. S. U. Chambéry)
M.	JULLIEN Pierre	Mathématiques Pures
Mme.	RINAUDO Marguerite	Chimie
MM.	BLIMAN Samuel	E. I. E.
	BEGUIN Claude	Chimie Organique
	NEGRE Robert	I. U. T.

MAITRES DE CONFERENCES ASSOCIES

MM.	YAMADA Osamu	Physique du Solide
	NAGAO Makoto	Mathématiques Appliquées
	MAREZIO Massimo	Physique du Solide
	CHEECKE John	Thermodynamique
	BOUDOURIS Georges	Radioélectricité
	ROZMARIN Georges	Chimie Papetière

Ce travail a été réalisé sous la direction de Monsieur LAURENT, Maître de Conférences. Je tiens à lui exprimer ma plus vive reconnaissance et mon admiration. C'est lui qui m'a donné l'idée de ce travail ainsi que de nombreux conseils qui m'ont été précieux.

Je remercie Monsieur le Professeur KUNTZMANN de m'avoir fait l'honneur de présider le jury et Messieurs les Professeurs GASTINEL et BARRA qui ont accepté d'en faire partie.

Je félicite le secrétariat pour la rapidité avec laquelle il a assuré la frappe impeccable de cet ouvrage.

Je remercie tous ceux, qui, de près ou de loin, ont participé à la réalisation matérielle de ce travail, en particulier Monsieur MOUNET qui en a réalisé l'impression.

INTRODUCTION



Considérons le problème suivant : étant donnés n abscisses $x_1 < x_2 < \dots < x_n$ d'un intervalle $[a, b]$ et n nombres réels r_1, \dots, r_n , on veut trouver une fonction f "assez lisse" vérifiant :

$$f(x_i) = r_i, \quad i = 1, \dots, n$$

On peut chercher, parmi les fonctions ayant une dérivée seconde de carré sommable, la fonction f qui minimise :

$$\int_a^b |f''|^2 dt.$$

Ce problème admet une solution unique f appelée fonction "spline" d'interpolation. f est formée de morceaux de polynômes de degré 3 se raccordant aux x_i ainsi que leurs dérivées première et seconde [cf Schoenberg [23]].

Les données r_i peuvent provenir de résultats expérimentaux et être entachées d'erreur. Une première façon de tenir compte d'une incertitude sur les données consiste à faire un compromis entre le "lissage" et "l'approximation des données". On cherche alors la fonction f qui minimise :

$$\int_a^b |f''|^2 dt + \rho \sum_{i=1}^n (f(x_i) - r_i)^2 \quad (\rho > 0) \quad (\text{voir Schoenberg, Laurent [1]})$$

Une telle fonction est appelée fonction "spline" d'ajustement.

Si on peut évaluer l'erreur commise sur les données, on peut chercher f vérifiant :

$$f(x_i) \in [r_i - \varepsilon_i, r_i + \varepsilon_i], \quad i = 1, \dots, n$$

et minimisant $\int_a^b |f''|^2 dt.$

T A B L E D E S M A T I E R E S



INTRODUCTION

CHAPITRE I -FONCTIONS "SPLINE" RELATIVES A UN CONVEXE-

- 1 Définition
- 2 Caractérisation
- 3 Unicité

PREMIERE PARTIE : METHODES DIRECTES.

CHAPITRE II-FONCTIONS "SPLINE" RELATIVES A UN CONVEXE ET PROGRAMMATION QUADRATIQUE-

- 1 Définition d'un problème (P1) équivalent au problème (P)
- 2 Etude du problème (P1)
- 3 Principe des méthodes directes de résolution pour le problème (P1).

CHAPITRE III -METHODE DU GRADIENT PROJETE-

- 1 Définition du problème (P1').
- 2 Algorithme n°1 (gradient projeté)
- 3 Convergence de l'algorithme n°1
- 4 Procédures Algol

CHAPITRE IV -METHODES DE RELAXATION-

- 1 Principe des méthodes de relaxation
- 2 Algorithme n°2 (cf Méthode de Southwell)
- 3 Algorithme n°3 (cf Méthode de Gauss Seidel)
- 4 Procédures Algol.

CHAPITRE V -METHODES DE FRANCK ET WOLFE-

- 1 description des algorithmes n°4 et 4'
- 2 Convergence
- 3 Mise en oeuvre des algorithmes dans le cas du problème (P1) et procédures Algol.

CHAPITRE VI -METHODES DES DIRECTIONS CONJUGUEES-

- 1 Directions conjuguées
- 2 Algorithme n°5
- 3 Aspects pratiques de l'algorithme n°5 et procédures Algol

CHAPITRE VII -EXEMPLES NUMERIQUES ET COMPARAISON DES RESULTATS-

- 1 Etude théorique des exemples
- 2 Exemples numériques
- 3 Comparaison des méthodes

DEUXIEME PARTIE : METHODE DUALE

CHAPITRE VIII -ETUDE GEOMETRIQUE DU PROBLEME DES FONCTIONS SPLINE DANS UN CONVEXE-

- 1 Propriétés de sous ensembles particuliers de L
- 2 Caractérisation de $T(C) = \Omega$
- 3 Formule de dualité pour le problème (P).

CHAPITRE IX -ALGORITHME DUAL-

- 1 Description de l'algorithme dual
- 2 Convergence
- 3 Détermination analytique de l'itéré y^{v+1}
- 4 Exemples Numériques et procédure Algol

ANNEXE : Procédures Algol

Références

E R R A T A

- Introduction 1 3 au lieu de intersection finie d'hyperplans lire intersection finie de demi-espaces fermés.
- Page I-6 1 1 au lieu de (43) lire (H3)
- Page I-6 1 9 au lieu de $(\ell, x)_X$ lire $(\ell | x)_X$
- Page I-6 1 16 lire $T^*T\sigma + \sum_{\ell \in L_1} \lambda_\ell \ell + \sum_{\ell \in L_2} \mu_\ell \ell = 0$
- Page I-8 1 19 au lieu de $\sum_{\ell \in L} \lambda_\ell \ell$ lire $\sum_{\ell \in L} \lambda_\ell \ell$
- Page II-2 1 4 lire $P = \{y \in \mathbb{R}^n \mid \sum_{i=1}^n \alpha_j^i y_i \leq c_j, j = 1, \dots, m\}$
- Page II-2 1 15 au lieu de relative à \mathbb{C} lire relative à $\mathbb{K} \frac{1}{y}$
- Page II-2 1 22 au lieu de $\sum_{i=1}^n b_j^i$ lire $\sum_{i=1}^n b_j^i k_i$
- Page II-5 1 10 lire $Q_y = \{y \in \mathbb{R}^n \mid F(y) < F(\bar{y})\}$
- Page II-5 1 15 lire $\Gamma(Q_{\bar{y}}, \bar{y}) =$
- Page II-5 1 17 au lieu de $F(y) - F(\bar{y})$ lire $F(y) < F(\bar{y})$
au lieu de $F(y) - F(\bar{y})$ lire $F(y) < F(\bar{y})$
- Page II-6 1 7 lire $I(\bar{y}) = \{j \in \{1, \dots, m\} \mid \sum_{i=1}^n \alpha_j^i \bar{y}_i = c_j\}$.
- Page III-3 avant dernière ligne lire $F(y^{v+1}) - F(y^v) \leq -\delta \|s^v\| < 0$
- Page IV-3 1 8 au lieu de s^v lire \bar{s}
- Page IV-5 1 1 lire GAUSS-SEIDEL
- Page IV-5 1 8 lire $\lambda^i = \lambda(x^{v,i}, s_i) = \min(\lambda'(x^{v,i}, s_i), \lambda''(x^{v,i}, s_i))$
- Page IV-5 1 15 au lieu de $=$ lire \subset
- Page IV-6 1 9 au lieu de $F^X(x^{i+1})$ lire $F(x^{i+1})$
- Page V-1 1 8 & 9 entre hyperplan et pour $K \cap P$ ajouter d'appui.
- Page V-5 1 13 au lieu de $\rho < \tau$ lire $|\rho^v| < \tau$
- Page VI-1 1 2 lire $((\dagger | s^k)) = ((s | s^k)) - \mu_k((s^k | s^k))$ pour tout $k \in \{1, \dots, v-1\}$
- Page VI-5 1 8 au lieu de $\Gamma^*(Q_{x^v}, x^v)$ lire $\Gamma(Q_{x^v}, x^v)$

Page VI-6 1 12 au lieu de Tenau [20] lire Temam [20]

Page VI-7 1 4 au lieu de $\geq \Delta$ lire ≥ 0

Page VI-7 1 9 au lieu de m_j^l lire m_i^l

Page VI-7 1 11 au lieu de $m_{i^v}^l$ lire m_i^l

Page VI-8 1 20 ajouter EP entre TØ, et EPS

Page VIII-2 1 16 au lieu de $(\ell_{ij} | \omega)_X$ lire $(\ell_{ij} | \omega_k)_X$

Page VIII-3 1 14 lire $i \in [1, \dots, m]$.

Page VIII-4 1 1 lire (T) étant de dimension q,

Page VIII-8 1 10 lire $C \cap \{x \in X | (\ell|x)_X > \alpha - \epsilon\} = \emptyset$

Page VIII-9 1 9 dernière ligne lire propriétés de \mathcal{Y} et la caractérisation du convexe T(C)

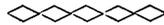
Page IX-4 1 7 lire D_γ est un demi-espace fermé contenant Ω

Page IX-5 1 au lieu de et $N(T)^\perp$ lire de $N(T)^\perp$

Page IX-13 1 3 au lieu de $\sum_{j \in H} m_j$ lire $\sum_{j \in H} |m_j|$

ANNEXE V 1 1 & 2 lire PROCEDURE SPLUN(N, Y, MU) ;
ARRAY Y, MU ;

C H A P I T R E I



FONCTIONS "SPLINE" RELATIVES A UN CONVEXE.

Dans un premier paragraphe nous rappelons la définition des fonctions "spline" relatives à un convexe donné sous la forme d'une intersection finie ~~de demi-espaces~~ ^{fermé} ~~d'hyperplans~~. Nous montrons l'existence de telles fonctions "spline" et, dans un deuxième paragraphe, nous en donnons une caractérisation. Le troisième paragraphe est consacré à l'étude des propriétés d'unicité.

I-DEFINITION-

Soient X et Y deux espaces de Hilbert définis sur \mathbb{R} et T une application linéaire et continue de X sur Y. Le noyau de T, noté $\mathcal{N}(T)$, est supposé de dimension q.

Soit L une partie finie de X et C le convexe fermé de X défini par :

$$C = \{ x \in X \mid \forall \ell \in L \quad (\ell | x)_X \leq a(\ell) \}$$

où les $a(\ell)$, $\ell \in L$, sont des nombres réels donnés.

C est supposé non vide.

Problème (P)

"Existe-t-il un élément σ de C qui vérifie :

$$\|T\sigma\|_Y = \min (\|Tx\|_Y \mid x \in C) ? "$$

Un tel élément, s'il existe, sera appelé "fonction "spline" relative à C et à T".

Proposition 1.1. (théorème d'existence).

"Le problème (P) a au moins une solution".

Démonstration :

a) C est convexe et T est linéaire donc T(C) est convexe.

b) T(C) est fermé.

Une condition nécessaire et suffisante pour que T(C) soit fermé est que $C + \mathcal{N}(T)$ soit fermé (cf Attéia [4]). Nous allons donc montrer que si $\lim_{n \rightarrow \infty} x_n + y_n = z$ avec $x_n \in C$, $y_n \in \mathcal{N}(T)$, $z \in X$, alors $z \in C + \mathcal{N}(T)$.

Soit K le sous-espace vectoriel engendré par L. K est supposé de dimension n. Si $\{ k_i, i = 1, \dots, n \}$ est une base de K, posons :

$$\forall \ell \in L \quad \ell = \sum_{i=1}^n \alpha_{\ell i} k_i \quad (\alpha_{\ell i} \in \mathbb{R} \quad \forall i = 1, \dots, n).$$

Considérons \mathbb{R}^n muni de la norme euclidienne, notée $\| \cdot \|$, et l'application A, linéaire et continue, de X dans \mathbb{R}^n définie par :

$$A(x) = [(k_1 | x)_X, \dots, (k_n | x)_X].$$

Posons $P = \{ y \in \mathbb{R}^n \mid \forall \ell \in L \quad \sum_{i=1}^n \alpha_{\ell i} y_i \leq a(\ell) \}$.

Lemme 1.1. " $A^{-1}(P) = C$ et $A(C) = P$."

Démonstration :

L'inclusion $A(C) \subset P$ est évidente et entraîne que $A^{-1}(P) \supset C$. Montrons que $A^{-1}(P) \subset C$. Soit $x \in A^{-1}(P)$ et $y = A(x)$. Alors $y \in P$, donc :

$$\forall \ell \in L \quad \sum_{i=1}^n \alpha_{\ell i} (k_i | x)_X = \sum_{i=1}^n \alpha_{\ell i} y_i \leq a(\ell)$$

et $x \in C$. Montrons maintenant que $A(C) \supset P$. Soit $y \in P$. Comme le noyau de A est K^\perp , l'orthogonal de K dans X, la restriction de A à K est une application

bijjective de K dans \mathbb{R}^n . Il existe donc $x \in K$ vérifiant $Ax = y$. On montre alors, comme précédemment, que $x \in C$ et $y \in A(C)$.

$A(C)$ est donc un polyèdre convexe de \mathbb{R}^n . Pour tout n , $A(x_n) \in P$ et $A(z - y_n) \in A(z + \mathcal{N}(T))$ qui est une variété linéaire de \mathbb{R}^n . Comme A est linéaire et continue,

$$\lim_{n \rightarrow \infty} \|z - (x_n + y_n)\|_X = 0 \text{ donc } \lim_{n \rightarrow \infty} \|A(z - y_n) - A(x_n)\|_X = 0.$$

et ceci entraîne que $P \cap A(z + \mathcal{N}(T))$ n'est pas vide. Il existe donc $y \in \mathcal{N}(T)$ tel que $A(z-y) \in P$. Comme $C = A^{-1}(P)$, il existe $x = z-y \in C$. Donc $z \in C + \mathcal{N}(T)$.

c) $T(C)$ étant convexe et fermé, la projection z de l'origine de Y sur $T(C)$ est l'élément unique qui vérifie :

$$\|z\|_Y = \min(\|y\|_Y \mid y \in T(C)) = \min(\|Tx\|_Y \mid x \in C).$$

$T^{-1}(z) \cap C$ est un convexe non vide de X . C' est l'ensemble des solutions de P .

Remarque :

Si σ_1 et σ_2 sont deux fonctions "spline" relatives à C et T , on voit donc que $\sigma_1 - \sigma_2 \in \mathcal{N}(T)$.

II-CARACTERISATION DES FONCTIONS "SPLINE" RELATIVES A C ET A T.-

Sans restreindre la généralité du problème, nous pouvons faire l'hypothèse suivante :

$$(H1) \quad " \forall l \in L, L \cap \{ \lambda l \mid \lambda \in \mathbb{R} \} \subset \{ -l \} \cup \{ +l \} " .$$

Faisons maintenant les hypothèses;

$$(H2) \quad " C \cap \mathcal{N}(T) = \emptyset " ,$$

$$(H3) \quad " I = \{ x \in X \mid \forall l \in L \quad (l \mid x)_X < a(l) \} \neq \emptyset . "$$

Remarque :

L'hypothèse (H3) entraîne alors que l'intérieur de C est I.

Proposition 1.2. (théorème de caractérisation).

"Une condition nécessaire et suffisante pour que $\sigma \in C$ soit une fonction "spline" relative à C et à T est qu'il existe des coefficients positifs ou nuls, $\lambda_\ell, \ell \in L$, tels que :

$$T^*T\sigma + \sum_{\ell \in L} \lambda_\ell \ell = 0$$

avec $\lambda_\ell = 0$ si $(\ell | \sigma)_X < a(\ell)$, où T^* désigne l'opérateur adjoint de T. "

Remarque :

T^* est une application linéaire et continue de Y sur l'orthogonal du noyau de T, noté $\mathcal{N}(T)^\perp$.

Démonstration :

Nous n'en rappelons ici que les grandes lignes (cf Laurent [15]).

Soit $x_0 \in X$ et E une partie de X. On appelle "déplacement intérieur" (pour E à partir de x_0) tout vecteur non nul $x \in E$ tel qu'il existe un voisinage V de x et un réel $\varepsilon > 0$ tel que :

$$\forall \eta \in]0, \varepsilon[, \forall y \in V \implies x_0 + \eta y \in E.$$

On peut vérifier que l'ensemble des déplacements intérieurs issus de x_0 est un cône ouvert de sommet l'origine que l'on notera $\Gamma(E, x_0)$.

Posons $B = \{x \in X \mid \|Tx\|_Y < \|T\sigma\|_Y\}$. Grâce à l'hypothèse (H2), B est un ouvert non vide. Soit $\Gamma(C, \sigma)$ (respectivement $\Gamma(B, \sigma)$) le cône des déplacements intérieurs pour C (resp. pour B) à partir de σ . Comme B et C sont d'intérieur non vide, ces cônes ne sont pas vides et on a le résultat suivant :

Proposition 1.3

"Une condition nécessaire et suffisante pour que $B \cap C = \emptyset$, c'est-à-dire pour que σ soit solution du problème (P) est que l'on ait :

$$\Gamma(B, \sigma) \cap \Gamma(C, \sigma) = \emptyset."$$

D'autre part, on a le théorème de séparation suivant :

Proposition 1.4

"Une condition nécessaire et suffisante pour que $\Gamma(B, \sigma) \cap \Gamma(C, \sigma) = \emptyset$ est qu'il existe $l_0, l_1 \in X$ tels que :

$$\underline{l_0 \neq 0 \quad l_0 + l_1 = 0}$$

$$\underline{\forall x \in \Gamma(B, \sigma) \quad , \quad (l_0 | x)_X \geq 0}$$

$$\underline{\forall x \in \Gamma(C, \sigma) \quad , \quad (l_1, x)_X \geq 0 \quad "}$$

On peut montrer que :

$$\Gamma(B, \sigma) = \{x \in X \mid (T^*T\sigma | x)_X < 0\}.$$

On en déduit - $l_0 = \lambda_0 T^*T\sigma$ avec $\lambda_0 > 0$.

On a par ailleurs :

$$\Gamma(C, \sigma) = \{x \in X \mid \forall l \in F(\sigma) \quad , \quad (l | x) < 0\}$$

$$\text{avec } F(\sigma) = \{l \in L \mid (l | \sigma)_X = a(l)\}.$$

On en déduit que $-l_1 \in CC(F(\sigma))$ où $CC(F(\sigma))$ désigne le plus petit cône convexe contenant $F(\sigma)$ (qui est fermé puisque $F(\sigma)$ est fini).

La proposition 1.4 peut alors s'énoncer :

"Une condition nécessaire et suffisante pour que $\Gamma(B, \sigma) \cap \Gamma(C, \sigma) = \emptyset$ est que :

$$\underline{-T^*T\sigma \in CC(F(\sigma))}"$$

et l'égalité :

$$CC(F(\sigma)) = \left\{ \sum_{l \in F(\sigma)} \lambda_l l \mid \lambda_l \geq 0 \right\}$$

entraîne alors le théorème de caractérisation.

Il peut arriver que la condition (43) ne soit pas vérifiée, ce qui est le cas, par exemple, si \mathcal{C} est défini par des "conditions d'interpolation". Supposons que l'on ait une partition de L en L_1, L_2, L_3 telle que $L_2 = -L_3$ et pour tout $l \in L_2$ alors $a(-l) = -a(l)$.

Posons :

$$C_1 = \{x \in X \mid \forall l \in L_1, (l \mid x)_X \leq a(l)\}$$

$$C_2 = \{x \in X \mid \forall l \in L_2, (l \mid x)_X = a(l)\}.$$

C_2 est une variété linéaire parallèle au sous-espace vectoriel V défini par :

$$V = \{x \in X \mid \forall l \in L_2, (l, x) = 0\}.$$

Supposons l'hypothèse suivante vérifiée :

$$(H3') \quad "I_1 = \{x \in X \mid \forall l \in L_1, (l \mid x)_X < a(l)\} \neq \emptyset".$$

On a alors le théorème de caractérisation suivant :

Proposition 1.5.

"Une condition nécessaire et suffisante pour que σ soit solution du problème (P) est qu'il existe $\lambda_l \geq 0, l \in L_1$ et $\mu_l (l \in L_2)$ tels que :

$$\underline{T^* T\sigma + \sum_{l \in L_1} \lambda_l l + \sum_{l \in L_2} \mu_l l}$$

avec

$$\underline{\lambda_l = 0 \text{ si } (l \mid \sigma)_X < a(l)}$$

Démonstration.

Soit $x_0 \in X$ et E une partie de X . On appelle "déplacement adhérent" (pour E à partir de x_0) tout vecteur $x \in E$ tel que pour tout voisinage V de x et tout $\varepsilon > 0$ il existe $\eta \in]0, \varepsilon[$ et $y \in V$ tels que :

$$x_0 + \eta y \in E.$$

On notera $\Gamma^*(E, x_0)$ le cône fermé des déplacements adhérents issus de x_0 .

Soit $\Gamma^*(C_2, \sigma)$ le cône des déplacements adhérents pour C_2 à partir de σ . On a alors le théorème suivant :

Proposition 1.6

"Une condition nécessaire et suffisante pour que $B \cap (C_1 \cap C_2) = \emptyset$ est que l'on ait :

$$\underline{\Gamma(B, \sigma) \cap \Gamma(C_1, \sigma) \cap \Gamma^*(C_2, \sigma) = \emptyset}.$$

On a d'autre part le résultat suivant :

Proposition 1.7. (Dubovickii et Miljutin [9], Lobry [21])

" Une condition nécessaire et suffisante pour que $\Gamma(B, \sigma) \cap \Gamma(C_1, \sigma) \cap \Gamma^*(C_2, \sigma) = \emptyset$ est qu'il existe $l_0, l_1, l_2 \in X$ tels que :

$$l_0 \neq 0, \quad l_0 + l_1 + l_2 = 0$$

$$\underline{Vx \in \Gamma(B, \sigma) \quad (l_0 | x)_X \geq 0,}$$

$$\underline{Vx \in \Gamma(C_1, \sigma) \quad (l_1 | x)_X \geq 0,}$$

$$\underline{Vx \in \Gamma^*(C_2, \sigma) \quad (l_2 | x)_X \geq 0.}"$$

On peut montrer que $\Gamma^*(C_2, \sigma) = V$. Les propositions 1.6 et 1.7 peuvent alors s'énoncer :

Une condition nécessaire et suffisante pour que $B \cap (C_1 \cap C_2) = \emptyset$ est que :

$$\underline{-T^* T \sigma \in CC(F_1(\sigma)) + V^\perp}$$

avec

$$\underline{F_1(\sigma) = \{ l \in L_1 \mid (l | x)_X = a(l) \} .}$$

d'où le théorème de caractérisation.

III-UNICITE-

Ajoutons maintenant l'hypothèse suivante :

(H4) "Quels que soient l_1, \dots, l_q , q éléments distincts et non opposés de L ($l_i \neq \pm l_j$ si $j \neq i$) nous avons :

$$(l_i | p)_X = 0 \quad , \quad i = 1, \dots, q \text{ et } p \in \mathcal{P}(T) \iff p = 0" ;$$

ce qui est équivalent à :

" $\det[(l_i | \omega_j)_X] \neq 0$ si $\{\omega_i, i = 1, \dots, q\}$ est une base de $\mathcal{P}(T)$ " .

Proposition 1.8 (Théorème d'unicité).

"Si les hypothèses (H1), (H2), (H3 ou H3'), et (H4) sont vérifiées alors le problème P admet une solution unique".

Démonstration.

Soient σ_1 et σ_2 deux solutions de (P). D'après la remarque faite au paragraphe 1, $\sigma_1 - \sigma_2 \in \mathcal{P}(T)$ et, comme l'ensemble des solutions de (P) est convexe, $\sigma = \frac{\sigma_1 + \sigma_2}{2}$ est encore solution de (P). Alors $F(\sigma) \subset F(\sigma_1) \cap F(\sigma_2)$, où $F(\sigma)$ est défini comme dans la démonstration de la proposition 1.2.: en effet, si $(l | x)_X = a(l)$ comme $(l | \sigma_1)_X \leq a(l)$ et $(l | \sigma_2)_X \leq a(l)$, cela entraîne que :

$$(l | \sigma)_X = (l | \sigma_1)_X = (l | \sigma_2)_X .$$

Par conséquent, si $l \in F(\sigma)$ alors $(l | \sigma_1 - \sigma_2)_X = 0$. D'autre part, d'après le théorème de caractérisation pour la solution σ (prop. 1.2 ou 1.5) il existe des coefficients λ_l non tous nuls tels que ${}^*T\sigma + \sum_{l \in L} \lambda_l l = 0$ avec $\lambda(l) = 0$ si $l \notin F(\sigma)$. On a donc $\sum_{l \in L} \lambda_l l \in \mathcal{P}(T)^\perp$ et l'hypothèse (H4) entraîne alors que $F(\sigma)$ contient au moins $q+1$ éléments distincts et non opposés. Comme $(l | \sigma_1 - \sigma_2)_X = 0$ pour tout $l \in F(\sigma)$, l'hypothèse (H4) entraîne que $\sigma_1 - \sigma_2 = 0$; d'où l'unicité.

Remarque

Dans le cas où l'on a des conditions d'interpolation avec l'hypothèse (H3'), l'hypothèse suivante, plus simple à vérifier que l'hypothèse H4, entraîne aussi l'unicité :

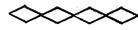
$$(H4') \quad " \forall l \in L_2 \quad (l p)_X = 0 \text{ avec } p \in \mathcal{D}(T) \Leftrightarrow p = 0$$

Les hypothèses (H1), (H2), (H3) ou (H3'), (H4) étant vérifiées, nous nous proposons d'obtenir la fonction "Spline" relative à C et à T au moyen d'un algorithme itératif, c'est-à-dire de construire une suite $\{x_n\} \in X$ telle que :

$$\underline{\lim_{x \rightarrow \infty} x_n = \sigma .}$$

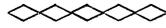
P R E M I E R E

P A R T I E



M E T H O D E S D I R E C T E S

C H A P I T R E II



FONCTIONS "SPLINE" RELATIVES A UN CONVEXE ET PROGRAMMATION QUADRATIQUE

Dans le paragraphe 1 nous montrons que le problème (P) est équivalent à un problème (P1) de programmation quadratique consistant à minimiser une forme quadratique "dégénérée" sur un polyèdre convexe. Dans le paragraphe 2 nous étudions les propriétés de cette forme quadratique et nous voyons que l'on peut retrouver simplement les théorèmes d'existence, de caractérisation et d'unicité énoncés dans le chapitre précédent. Dans le paragraphe 3, enfin, nous donnons les principes généraux sur lesquels seront fondés les méthodes "directes" permettant de calculer la solution du problème (P) au moyen d'un algorithme calculant la solution du problème (P1).

I-DEFINITION D'UN PROBLEME EQUIVALENT AU PROBLEME (P).-

Nous reprenons la définition introduite au cours de la démonstration de la proposition 1.1.

Soit K le sous-espace engendré par L. K est supposé de dimension n ($n > q$).

Soit $\{k_1, \dots, k_n\}$ une base de K.

Soit $L = \{l_1, \dots, l_m\}$ et posons $a(l_j) = c_j$.

$C = \{x \in X \mid (l_i, x)_X \leq c_j \quad j = 1, \dots, m\}$.

Les éléments de L peuvent être représentés par leurs composantes sur la base,

c'est-à-dire $l_j = \sum_{i=1}^n \alpha_j^i k_i \quad \forall j \in \{1, \dots, m\}$

Considérons \mathbb{R}^n muni de la norme euclidienne (on notera $\|\cdot\|$ la norme

dans \mathbb{R}^n et $(\cdot | \cdot)$ le produit scalaire) et A l'application linéaire et continue de X dans \mathbb{R}^n définie par :

$$A(x) = [(k_1 | x)_X, \dots, (k_n | x)_X].$$

Posons $P = \{y \in \mathbb{R}^n \mid \sum_{i=1}^n \alpha_j^i y_i \leq c_j, j=1, \dots, m\}$. P est un polyèdre convexe de \mathbb{R}^n .

Soit $y \in \mathbb{R}^n$ et K_y^\perp la variété linéaire de X définie par :

$$\begin{aligned} K_y^\perp &= \{x \in X \mid A(x) = y\} \\ &= \{x \in X \mid (k_i | x)_X = y_i, i = 1, \dots, n\} \\ &= \{x \in X \mid (\sum_{i=1}^n \alpha_j^i k_i | x)_X = \sum_{i=1}^n \alpha_j^i y_i, j = 1, \dots, m\} \end{aligned}$$

Si $y \in P$ alors $K_y^\perp \subset C$. Si l'hypothèse (H2) est vérifiée pour C elle est vérifiée pour K_y^\perp dès que $y \in P$. Nous supposons dans toute la suite de ce chapitre que les hypothèses (H1), (H2), (H3) ou (H'3) et (H4) sont vérifiées. La proposition 1.8 entraîne alors :

" si $y \in P$, il existe un élément unique de K_y^\perp , noté σ_y , qui vérifie :

$$\|\sigma_y\|_Y = \min (\|Tx\|_Y \mid x \in K_y^\perp). \sigma_y \text{ est la fonction "spline" relative à } C \text{ et à } T.$$

Des théorèmes dûs à Laurent et à Anselone [1] permettent de préciser ce qu'est l'élément σ_y de Y .

Soit $\{h_j, j = 1, \dots, n-q\}$ une base de $K \cap \mathcal{N}(T)$, sous espace vectoriel de dimension $n-q$ représenté par ses composantes sur la base de $K \{k_i, i = 1, \dots, n\}$ c'est-à-dire :

$$(2.1) \quad h_j = \sum_{i=1}^n b_j^i k_i \quad \forall j \in \{1, \dots, n-q\}$$

On pose $T^{*-1}(h_j) = \varphi_j \quad \forall j \in \{1, \dots, n-q\}$. Et si les μ_j vérifient :

$$(2-2) \quad \sum_{j=1}^{n-q} \mu_j (\varphi_j | \varphi_i)_Y = \sum_{r=1}^r b_i^r y_r \quad \forall i \in \{1, \dots, n-q\}$$

alors on a les égalités :

$$(2-3) \quad T_{\sigma_y} = \sum_{j=1}^{n-q} \mu_j \varphi_j \quad \text{et,}$$

$$(2-4) \quad T^* T_{\sigma_y} = \sum_{i=1}^n \left(\sum_{j=1}^{n-q} b_j^i \mu_j \right) k_i.$$

Considérons les matrices ϕ et B définies par :

$$\phi_{ij} = (\varphi_i | \varphi_j)_Y \quad (i = 1, \dots, n-q)(j = 1, \dots, n-q)$$

$$\text{et } B_{ij} = b_i^j \quad (i = 1, \dots, n-q)(j = 1, \dots, n)$$

alors :

$$\|T_{\sigma_y}\|^2 = (By)^T \phi^{-1} By = y^T \Omega y = (y | \Omega y)$$

si l'on pose $\Omega = B^T \phi^{-1} B$.

Problème (P1).

"Existe-t-il $y^* \in P$ qui vérifie :

$$y^{*T} \Omega y^* = \min(y^T \Omega y \mid y \in P) ? "$$

Proposition 2.1.

" Les problèmes (P) et (P1) sont "équivalents". "

D'après le lemme 1.1, $A(C) = P$. Si x^* est solution du problème (P) alors $y^* = A(x^*)$ est solution de (P1) et, réciproquement, si y^* est solution de (P1) alors σ_y^* , la fonction "spline" relative à K^{\perp} et à T , est solution de (P) et l'on a $A(\sigma_y^*) = y^*$.

Remarque :

Etant donnée la définition de ϕ , la matrice ϕ^{-1} est définie positive; par conséquent, Ω est définie non négative. La fonctionnelle qui à $y \in \mathbb{R}^n$ associe le réel

$y^T \Omega y$ est donc une forme quadratique définie non négative. Le problème (P1) consiste donc à minimiser une forme quadratique définie non négative sur un polyèdre convexe de \mathbb{R}^n .

II-ETUDE DU PROBLEME (P1).-

2.1. Propriétés de la forme quadratique

Notations.

On appellera F la fonctionnelle qui à $y \in \mathbb{R}^n$ associe $y^T \Omega y$.

On posera $\mathcal{P}(B) = \{y \in \mathbb{R}^n \mid \sum_{i=1}^n b_j^i y_i = 0 \quad j = 1, \dots, n-q\}$.

On remarque que $\mathcal{P}(B)$ est de dimension q . On a d'autre part $A(\mathcal{P}(T)) = \mathcal{P}(B)$ (cf Laurent et Anselone [1]).

Comme Ω est une matrice définie non négative, on peut montrer que F est une fonctionnelle convexe continue. Son gradient est l'application de \mathbb{R}^n dans lui-même qui à $y \in \mathbb{R}^n$ associe le vecteur $2\Omega y$. On a en outre les propriétés suivantes :

$$(2-5) \quad F(y) = 0 \iff y \in \mathcal{P}(B) \iff y = Ax \text{ avec } x \in \mathcal{P}(T).$$

(2-6) $\Omega y \in \mathcal{P}(B)^\perp$. En effet, $\Omega y \in \text{Range}(\Omega)$ et $\text{Range}(\Omega) = \mathcal{P}(\Omega^*)^\perp = \mathcal{P}(\Omega)^\perp$ (puisque $\Omega = \Omega^*$) = $\mathcal{P}(B)^\perp$.

$$(2-7) \quad F(y_1) - F(y_2) = 2(y_1 - y_2 | \Omega y_2) + F(y_1 - y_2).$$

2.2. Le problème (P1) admet au moins une solution.

Nous allons montrer directement cette propriété.

Soit $y \in P$; y peut se mettre sous la forme : $y = y_1 + y_2$, avec $y_1 \in \mathcal{P}(B)^\perp$ et $y_2 \in \mathcal{P}(B)$, et d'après les propriétés (2-5, 6 et 7) on a $F(y) = F(y_1)$. Soit Q la projection de P sur $\mathcal{P}(B)^\perp$. Q est un polyèdre convexe de $\mathcal{P}(B)^\perp$. La restriction de F à $\mathcal{P}(B)^\perp$ étant une forme quadratique définie positive, il existe un point $z \in Q$, unique qui vérifie :

$$F(z) = \min(F(x) \mid x \in Q).$$

L'ensemble $(z + \mathcal{P}(B)) \cap P$ est donc un convexe non vide et c'est l'ensemble des solutions de (P1).

2.3 Caractérisation des solutions de (P1) (cf Laurent [16]).

Soit $\bar{y} \in P$. On pose $Q_{\bar{y}} = \{y \in \mathbb{R}^n \mid F(y) < F(\bar{y})\}$. $Q_{\bar{y}}$ est un convexe ouvert. De plus l'hypothèse (H2) entraîne que $\mathcal{P}(B) \cap P = \emptyset$ donc $Q_{\bar{y}}$ est non vide. Soit $\Gamma(Q_{\bar{y}}, \bar{y})$ le cône des déplacements intérieurs pour $Q_{\bar{y}}$ à partir de \bar{y} (cf démonstration de la proposition 1.2) On peut montrer que :

$$\Gamma(Q_{\bar{y}}, \bar{y}) = \{s \in \mathbb{R}^n \mid s = \lambda(y - \bar{y}), \lambda > 0, y \in Q_{\bar{y}}\}.$$

Lemme 2.1 : " $\Gamma(Q_{\bar{y}}, \bar{y}) = \{s \in \mathbb{R}^n \mid (s \mid \Omega \bar{y}) < 0\}$."

Démonstration :

a) Soit $s = \lambda(y - \bar{y})$ avec $\lambda > 0$ et $y \in Q_{\bar{y}}$. On a $F(y) < F(\bar{y})$, donc la propriété (2-7) entraîne :

$F((y - \bar{y}) + \bar{y}) - F(\bar{y}) = 2(y - \bar{y} \mid \Omega \bar{y}) + F(y - \bar{y}) < 0$ donc $(y - \bar{y} \mid \Omega \bar{y}) < 0$ et par conséquent $(s \mid \Omega \bar{y}) < 0$

b) Soit s tel que $(s \mid \Omega \bar{y}) < 0$. Alors, pour λ assez petit, on a $\lambda s + \bar{y} \in Q_{\bar{y}}$. En effet :

$$F(\lambda s + \bar{y}) - F(\bar{y}) = \lambda[2(s \mid \Omega \bar{y}) + \lambda F(s)],$$

et comme $(s|\Omega y) \neq 0$ entraîne $F(s) \neq 0$, pour $\lambda \in]0, -2 \frac{(s|\Omega y)}{F(s)} [$
on a bien $F(\lambda s + \bar{y}) < F(\bar{y})$.

$P = \{y \mid \sum_{i=1}^n \alpha_j^i y_i \leq c_j, j = 1, \dots, m\}$ est un convexe fermé. Soit $\Gamma^*(P, \bar{y})$

le cône des déplacements adhérents pour P à partir de \bar{y} . On peut montrer que :

$$\Gamma^*(P, \bar{y}) = \text{adh} \{s \in \mathbb{R}^n \mid s = \lambda(y - \bar{y}), y > 0, y \in P\}.$$

Si on pose :

$$I(\bar{y}) = \{j \in [1, \dots, m] \mid \sum_{i=1}^n \alpha_j^i \bar{y}_i = c_j\}$$

On peut alors vérifier que :

$$\Gamma^*(P, \bar{y}) = \{s \in \mathbb{R}^n \mid \sum_{i=1}^n \alpha_j^i s_i \leq 0 \quad \forall j \in I(\bar{y})\}.$$

On a le résultat suivant :

Proposition 2.2.

"Une condition nécessaire et suffisante pour que l'on ait $Q_{\bar{y}} \cap P = \emptyset$, c'est-à-dire pour que \bar{y} soit solution du problème (P1) est que :

$$\underline{\Gamma(Q_{\bar{y}}, \bar{y}) \cap \Gamma^*(P, \bar{y}) = \emptyset}$$

et d'après le théorème de séparation pour n -cônes convexes, cité au chapitre précédent (proposition 1.7) (cf Dubovickii et Miljutin [9], Lobry [21], Laurent [16]) on peut caractériser les solutions de (P1) de la manière suivante :

Proposition 2.3.

"Une condition nécessaire et suffisante pour que \bar{y} soit solution du problème (P1) est qu'il existe des coefficients positifs $\lambda_j, j \in I(\bar{y})$, tels que :

$$\underline{\Omega y + \sum_{j \in I(\bar{y})} \lambda_j \alpha_j = 0}$$

Remarque.

Les relations (2-1) et (2-4) permettent d'écrire :

$$\begin{aligned}
 -T^* T \sigma_y &= \sum_{i=1}^n \left(\sum_{j=1}^{n-q} b_j^i \mu_j \right) k_i = \sum_{i=1}^n [(B^T \phi^{-1} B) y]_i k_i \\
 &= \sum_{i=1}^n (\Omega y)_i k_i
 \end{aligned}$$

donc $T^* T \sigma_y + \sum_{i \in I(y)} \lambda_j \sum_{j=1}^n \alpha_j^i k_i = 0$ et on retrouve le théorème de caractérisation énoncé au chapitre précédent.

2.4 Unicité de la solution.

La relation $\mathcal{P}(B) = A(\mathcal{P}(T))$ et l'hypothèse (H4) entraînent la propriété suivante :

"Quels que soient $\alpha_{j_1}, \dots, \alpha_{j_q}$ q éléments distincts et non opposés de l'ensemble

$\{\alpha_j, j=1, \dots, m\}$, nous avons :

$$(\alpha_{j_k} | y) = 0 \quad k = 1, \dots, q \text{ et } y \in \mathcal{P}(B) \iff y = 0"$$

ce qui équivaut à :

"Toute combinaison linéaire non nulle d'éléments de $\{\alpha_j, j = 1, \dots, m\}$ orthogonale à un élément de $\mathcal{P}(B)$ contient au moins $q+1$ éléments".

On retrouve alors directement l'unicité de la solution de (P1) : supposons que y_1 et y_2 soient deux solutions de (P1). Nous avons vu au paragraphe 2.2 de ce chapitre que $y_1 - y_2 \in \mathcal{P}(B)$. Comme l'ensemble des solutions de (P1) est convexe, $y = \frac{y_1 + y_2}{2}$ est solution. On vérifie que $I(y) \subset I(y_1) \cap I(y_2)$. L'hypothèse (H2) et la propriété $A^{-1}(P) = C$ entraînent : $P \cap \mathcal{P}(B) = \emptyset$ donc $\Omega y \neq 0$. La propriété ci-dessus entraîne alors que $I(y)$ contient au moins $q+1$ indices définissant $q+1$ éléments distincts et non opposés de $\{\alpha_j, j=1, \dots, m\}$. Et comme pour tout $j \in I(y)$,

$(\alpha_j \parallel y_2 - y_1) = 0$, cette propriété entraîne encore $y_2 - y_1 = 0$. D'où l'unicité de la solution de (P1) dont on déduit que (P) est une solution unique.

III-PRINCIPE DE METHODES DIRECTES DE RESOLUTION POUR LE PROBLEME (P1).

Soit y_σ la solution unique du problème (P1).
Posons $d = F(y_\sigma) = \min (F(y) = y^T \Omega y \mid y \in P)$.

3.1. Suites minimisantes pour F sur le domaine P.

On appelle "suite minimisante" (pour F sur le domaine P) toute suite $\{y^\nu\}$, $y^\nu \in P$, vérifiant :

$$\lim_{\nu \rightarrow \infty} F(y^\nu) = d.$$

Proposition 2.4.

"Toute suite minimisante est bornée".

Démonstration

Soient $\alpha > d$ et $P_\alpha = \{y \in P \mid F(y) < \alpha\}$. Alors $y_\sigma \in P_\alpha$. Si P_α n'était pas borné, il existerait un élément s de \mathbb{R}^n ($s \neq 0$) tel que :

$$\forall \lambda > 0 \quad \text{alors } y_\sigma + \lambda s \in P_\alpha.$$

Or $s \in \Gamma^*(P, y_\sigma)$ donc $s \in \Gamma(y_\sigma, y_\sigma)$ et on a $(s \mid \Omega y) \geq 0$.

L'égalité $F(y_\sigma + \lambda s) - F(y_\sigma) = 2 \lambda (s \mid \Omega y_\sigma) + \lambda^2 F(s) \neq 0$ entraîne que $F(y_\sigma + \lambda s)$ est une fonction croissante non bornée de λ sur $[0, \infty]$, ce qui est en contradiction avec l'inégalité $F(y_\sigma + \lambda s) < \alpha$. P_α est donc borné, ce dont il découle immédiatement que toute suite minimisante est bornée.

Proposition 2.5

" Toute suite minimisante converge vers la solution y_σ de (P1) ".

Démonstration

Soit $\{y^\nu\}$ une suite minimisante. Supposons qu'elle ne converge pas vers y_σ . On peut alors en extraire une sous-suite $\{y^{\nu_p}\}$ telle que :

$$\exists \varepsilon : \forall p \quad \| y^{\nu_p} - y_\sigma \| > \varepsilon .$$

D'après la proposition 2.3, la suite $\{y^{\nu_p}\}$ est bornée. On peut donc en extraire une sous-suite convergente $\{y^{\nu_q}\}$. Soit y la limite de cette suite. On a $\| y_\sigma - y \| > \varepsilon$.

D'autre part $F(y) = \lim_{q \rightarrow \infty} F(y^{\nu_q}) = \lim_{\nu \rightarrow \infty} F(y^\nu) = d$ et l'hypothèse d'unicité

entraîne : $y = y_\sigma$. D'où la contradiction.

Donc $\{y^\nu\}$ converge vers y_σ .

3.2. Principes généraux des méthodes "directes".

Les méthodes décrites dans les chapitres suivants seront des méthodes de construction des suites minimisantes.

Etant donné $y^\nu \in P$, $y^\nu \neq y_\sigma$, $y^{\nu+1}$ sera déterminé par le choix d'une direction s^ν , appelée "direction de descente", puis par le choix d'un "pas" λ^ν dans cette direction : $y^{\nu+1} = y^\nu + \lambda^\nu s^\nu$. On prendra toujours $s^\nu \in \Gamma(Q, y^\nu) \cap \Gamma^*(P, y^\nu)$. Nous verrons plus loin les différents choix possibles pour s^ν . Le pas λ^ν sera déterminé de la même façon dans toutes les méthodes.

Soient $\lambda'(y^\nu, s^\nu)$ et $\lambda''(y^\nu, s^\nu)$ définis par :

$$\lambda'(y^\nu, s^\nu) = \max(\lambda \in [0, \infty] \mid y^\nu + \lambda s^\nu \in P).$$

et $F(y^\nu + \lambda''(y^\nu, s^\nu)s^\nu) = \min(F(y^\nu + \lambda s^\nu) \mid \lambda \in [0, \infty])$.

Alors $\lambda^\nu = \min(\lambda'(y^\nu, s^\nu), \lambda''(y^\nu, s^\nu))$.

3.3 Méthodes directes et solution du problème (P).

Rappels (cf Atteia [2]).

Soit e_i le vecteur de \mathbb{R}^n dont les composantes vérifient :

$$e_i^j = 0 \quad \text{si } i \neq j, \quad e_i^i = 1.$$

et σ_i la fonction "spline" unique relative à $K_{e_i}^L$ et à T. Les σ_i , $i = 1, \dots, n$, sont linéairement indépendantes et, si σ_y est la fonction "spline" relative à K_y^L , on peut écrire

$$\sigma_y = \sum_{i=1}^n y_i \sigma_i.$$

Soit \tilde{A} l'application de \mathbb{R}^n dans X qui à $y \in \mathbb{R}^n$ associe la fonction "spline" relative à K_y^L et à T. On a alors le résultat suivant :

Lemme 2.2. " \tilde{A} est linéaire et continu ".

Ceci est immédiat puisque l'on peut écrire : $\sigma_y = \sum_{i=1}^n y_i \sigma_i$. \tilde{A} est une application linéaire de \mathbb{R}^n dans l'espace de dimension n engendré par $\{\sigma_i, i = 1, \dots, n\}$. \tilde{A} est donc continue.

On déduit immédiatement de ce lemme la proposition suivante :

Proposition 2.6

" Soit $\{y^\nu\}$ une suite de P qui converge vers y et $\{\sigma_{y^\nu}\}$ la suite de C qui lui est associée. Alors $\{\sigma_{y^\nu}\}$ converge dans C vers σ_y ".

Corollaire 2.1.

" Si $\{y^\nu\}$ une suite minimisante, alors $\{\sigma_{y^\nu}\}$ est une suite de C qui converge vers la solution σ du problème (P). "

C H A P I T R E III



METHODE DU GRADIENT PROJETE

Dans le paragraphe 1 nous définissons le problème particulier (P1') résolu par l'algorithme du gradient projeté décrit au paragraphe 2. Dans le paragraphe 3 nous montrons la convergence de cet algorithme. Le paragraphe 4 est consacré à la description d'une procédure ALGOL.

I-DEFINITION DU PROBLEME (P1')-

Soient k_1, \dots, k_n , n éléments linéairement indépendants de X et C le convexe défini par :

$$C = \{x \in X \mid \alpha_i \leq (k_i \mid x)_X \leq \beta_i, \quad i = 1, \dots, n\}$$

avec $\alpha_i \leq \beta_i$ pour tout i ($|\alpha_i|$ ou $|\beta_i|$ peut éventuellement être infini)

L'hypothèse (H1) est vérifiée. On suppose que C vérifie les hypothèses (H2), (H3) ou (H3') et (H4).

On considère le polyèdre convexe de \mathbb{R}^n :

$$P = \{y \in \mathbb{R}^n \mid \alpha_i \leq y_i \leq \beta_i, \quad i = 1, \dots, n\},$$

et on définit, comme au chapitre II, le problème (P1') équivalent au problème (P') et consistant à trouver l'unique minimum sur P de la forme quadratique $F(y) = F(y) = (y \mid \Omega y)$.

Remarque :

La particularité du problème (P1') réside dans le fait qu'il consiste à minimiser une forme quadratique dégénérée sur un "pavé" de \mathbb{R}^n .

Soit $\bar{y} \in P$. On pose

$$(3_1) \quad I(\bar{y}) = I_+(\bar{y}) \cup I_-(\bar{y})$$

$$\text{où } I_+(\bar{y}) = \{ i \in \{1, \dots, n\} \mid \bar{y}_i = \beta_i \}$$

$$\text{et } I_-(\bar{y}) = \{ i \in \{1, \dots, n\} \mid \bar{y}_i = \alpha_i \}$$

Remarque :

Si P est d'intérieur vide, c'est-à-dire, s'il existe $j \in \{1, \dots, n\}$ tel que $\alpha_j = \beta_j$, alors

$$I_+(y) \cap I_-(y) \supset \{ j \} \quad \forall y \in P.$$

Si on pose $J = \{ i \in \{1, \dots, n\} \mid \alpha_i = \beta_i \}$, alors :

$$J = I_+(y) \cap I_-(y) \quad \forall y \in P$$

On peut alors énoncer un théorème de caractérisation très simple qui se déduit immédiatement de la proposition 2.3 :

Proposition 3.1 :

"Si P est d'intérieur non vide, une condition nécessaire et suffisante pour que \bar{y} soit solution de (Pl') est que $\bar{z} = \Omega\bar{y}$ vérifie la propriété suivante :

$$(3_2) \quad \begin{array}{ll} \bar{z}_i \leq 0 & \forall i \in I_+(\bar{y}) \\ \hline \bar{z}_i \geq 0 & \forall i \in I_-(\bar{y}) \\ \hline \bar{z}_i = 0 & \forall i \notin I(\bar{y}). \quad " \end{array}$$

On obtient de même le théorème suivant :

Proposition 3.2.

"Si P est d'intérieur vide, une condition nécessaire et suffisante pour que $\bar{y} \in P$ soit solution du problème (Pl') est que $\bar{z} = \Omega\bar{y}$ vérifie :

$$(3_3) \quad \begin{array}{lll} \bar{z}_i = 0 & \text{si} & i \notin I(\bar{y}) \\ \hline \bar{z}_i \leq 0 & \text{si} & i \in I_+(\bar{y}) - J \\ \hline \bar{z}_i \geq 0 & \text{si} & i \in I_-(\bar{y}) - J'' \end{array}$$

II-ALGORITHME n°1 -(Gradient projeté).

L'idée principale de cet algorithme est due à Rosen [22]. Elle consiste à choisir comme direction de descente, à la $\nu^{i\text{ème}}$ itération, la projection de $(-\Omega y^\nu)$ sur le cône $\Gamma^*(P, y^\nu)$. Le nom de l'algorithme provient du fait que le gradient de F au point y^ν est le vecteur $2\Omega y^\nu$. En fait nous ne choisirons pas toujours une telle direction de descente afin d'assurer la "convergence de l'algorithme".

Afin de clarifier l'exposé nous allons présenter l'algorithme dans le cas où P est d'intérieur non vide, c'est-à-dire où $J = \emptyset$.

2.1. Itération.

Soit $y^\nu \in P$. On pose $z^\nu = \Omega y^\nu$ et on désigne $I(y^\nu)$, $I_+(y^\nu)$ et $I_-(y^\nu)$, respectivement, par I^ν , I_+^ν et I_-^ν .

Soit $Q_y^\nu = \{ y \in \mathbb{R}^n \mid F(y) < F(y^\nu) \}$. Si y^ν n'est pas solution, le cône $\Gamma(Q_y^\nu, y^\nu) \cap \Gamma^*(P, y^\nu)$ n'est pas vide et on va choisir $s^\nu \in \Gamma(Q_y^\nu, y^\nu) \cap \Gamma^*(P, y^\nu)$.

A. Choix de la direction de descente "normale".

Soit \tilde{s} la projection de $-\Omega y^\nu$ sur le cône $\Gamma^*(P, y^\nu)$. \tilde{s} est défini par :

$$(3.4) \quad \tilde{s}_i = -z_i^\nu \quad \text{si } i \notin I^\nu,$$

$$\tilde{s}_i = \min(-z_i^\nu, 0) \quad \text{si } i \in I_+^\nu$$

$$\text{et } \tilde{s}_i = \max(-z_i^\nu, 0) \quad \text{si } i \in I_-^\nu$$

\tilde{s} vérifie alors l'inégalité

$$(3.5) \quad (\tilde{s} \mid \Omega y^\nu) = -\|\tilde{s}\|^2 < 0$$

et le lemme 2.1 entraîne que $\tilde{s} \in \Gamma(Q_y^\nu, y^\nu)$.

Dans cette direction, le pas sera déterminé comme nous l'avions annoncé au paragraphe 3.2 du chapitre II :

$$\text{si } \lambda'(y^\nu, \tilde{s}) = \max (\lambda > 0 \mid y^\nu + \lambda \tilde{s} \in P)$$

$$\text{et } F(y^\nu + \lambda''(y^\nu, \tilde{s})) = \min(F(y^\nu + \lambda \tilde{s}) \mid \lambda > 0)$$

$$\text{alors } \lambda(y^\nu, \tilde{s}) = \min(\lambda'(y^\nu, \tilde{s}), \lambda''(y^\nu, \tilde{s})).$$

$$\text{et on pose : } \tilde{y} = y^\nu + \lambda(y^\nu, \tilde{s}) \tilde{s}.$$

B. Modification éventuelle de la direction de descente.

Si la direction de descente "normale" nous amène à faire un pas trop petit, à cause des contraintes, nous la modifions :

Soit δ un nombre réel positif "assez petit" donné. ($\delta < \min(\beta_i - \alpha_i \mid i \in J)$)

Si on a $\lambda(y^\nu, \tilde{s}) = \lambda'(y^\nu, \tilde{s})$ et $\|\tilde{y} - y^\nu\| < \delta$, on prend alors la direction de descente s^ν définie par :

$$(3.6) \quad s_j^\nu = -z_j^\nu$$

$$\text{et } s_i^\nu = 0 \text{ pour } i \neq j$$

où j est un indice n'appartenant pas à I^ν et vérifiant :

$$\text{si } s_j^\nu > 0 \text{ alors } \lambda'(y^\nu, s^\nu) = \frac{b_i - y_i^\nu}{s_i^\nu}$$

$$\text{si } s_j^\nu < 0 \text{ alors } \lambda'(y^\nu, s^\nu) = \frac{a_i - y_i^\nu}{s_i^\nu}.$$

Par définition on a encore $s^\nu \in \Gamma^*(P, y^\nu)$ et la propriété (3.5), qui vérifie \tilde{s} , est encore vérifiée par s^ν . On en déduit que $s^\nu \in \Gamma^*(P, \delta^\nu) \cap \Gamma(Q_y^\nu, y^\nu)$.

On choisit encore

$$\lambda(y^\nu, s^\nu) = \min(\lambda'(y^\nu, s^\nu), \lambda''(y^\nu, s^\nu))$$

$$\text{et on pose } y^{\nu+1} = y^\nu + \lambda(y^\nu, s^\nu) s^\nu.$$

Dans les autres cas, c'est-à-dire si $\lambda(y^{\nu}, \tilde{s}) = \lambda''(y^{\nu}, \tilde{s})$ ou si $\|\tilde{y} - y^{\nu}\| \geq \delta$, la direction de descente reste la direction "normale" et on pose :

$$s^{\nu} = \tilde{s} \quad \text{et} \quad y^{\nu} = \tilde{y}.$$

2.2. Propriétés de l'itéré (cf Rosen [22])

$$y^{\nu+1} = y^{\nu} + \lambda(y^{\nu}, s^{\nu}) s^{\nu}.$$

La propriété (2.7) permet d'écrire :

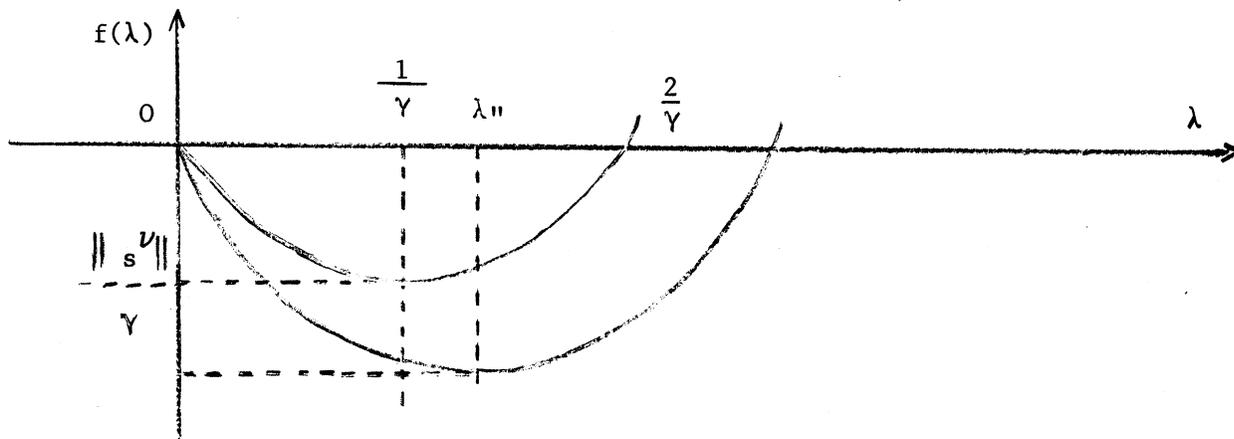
$$F(y^{\nu} + \lambda s^{\nu}) = F(y^{\nu}) - 2\lambda \|s^{\nu}\|^2 + \lambda^2 F(s^{\nu}) \quad \forall \lambda$$

Soit γ la plus grande valeur propre de Ω : γ est strictement positive et on a :

$$F(y^{\nu} + \lambda s^{\nu}) - F(y^{\nu}) \leq -\lambda(2 - \gamma\lambda) \|s^{\nu}\|^2.$$

On en déduit immédiatement :

$$(3.7) \quad F(y^{\nu} + \lambda s^{\nu}) < F(y^{\nu}) \quad \forall \lambda \in]0, \frac{2}{\gamma} [.$$



et l'inégalité :

$F(y^\nu + \lambda''(y^\nu, s^\nu) s^\nu) - F(y^\nu) \leq F(y^\nu + \frac{1}{\gamma} s^\nu) - F(y^\nu)$
 entraîne, si $\lambda(y^\nu, s^\nu) = \lambda''(y^\nu, s^\nu)$:

$$(3.8) F(y^{\nu+1}) - F(y^\nu) \leq -\frac{1}{\gamma} \|s^\nu\|^2 < 0$$

Si $\lambda(y^\nu, s^\nu) = \lambda'(y^\nu, s^\nu)$ alors

si l'on pose $f(\lambda) = F(y^\nu + \lambda s^\nu) - F(y^\nu)$, on a :

$$f'(\lambda'(y^\nu, s^\nu)) \leq 0 .$$

Or $f'(\lambda'(y^\nu, s^\nu)) = -2 \|s^\nu\|^2 + 2 \lambda'(y^\nu, s^\nu) F(s^\nu)$

donc $\lambda'(y^\nu, s^\nu) \leq \frac{\|s^\nu\|^2}{F(s^\nu)}$.

On a alors l'inégalité

$$F(y^{\nu+1}) - F(y^\nu) = -2 \lambda'(y^\nu, s^\nu) \|s^\nu\|^2 + \lambda'^2(y^\nu, s^\nu) F(s^\nu)$$

qui entraîne :

$$(3.9) F(y^{\nu+1}) - F(y^\nu) \leq -\lambda'(y^\nu, s^\nu) \|s^\nu\|^2 < 0$$

2.3 Cas où $J \neq \emptyset$

Théoriquement l'algorithme est inchangé : la direction de descente "normale" au point y^ν est encore la projection de $-\nabla y^\nu$ sur $\Gamma^*(P, y^\nu)$. Ceci consiste à choisir \tilde{s} tel que :

$$\tilde{s}_i = -z_i^\nu \quad \text{si } i \notin I^\nu ,$$

$$\tilde{s}_i = \min(-z_i^\nu, 0) \quad \text{si } i \in I_+^\nu - J$$

$$\tilde{s}_i = \max(-z_i^\nu, 0) \quad \text{si } i \in I_+^\nu - J$$

et $\tilde{s}_i = 0 \quad \text{si } i \in J.$

La modification éventuelle de la direction de descente se fait de la même façon que dans le cas où $J = \emptyset$.

III-CONVERGENCE DE L'ALGORITHME N°1.-

Supposons $J = \emptyset$. Nous démontrons d'abord le lemme suivant :

Lemme 3.1. "Soit $\{y^\nu\}$ une suite de P ayant les propriétés suivantes :

$$(3.10) \quad F(y^\nu) \leq F(y^{\nu-1}) \quad \forall \nu$$

$$(3.11) \quad \lim_{\nu \rightarrow \infty} \max(|z_i^\nu|, i \notin I^\nu) = 0$$

$$\lim_{\nu \rightarrow \infty} \max(0, -z_i^\nu \mid i \in I_-^\nu) = 0$$

$$\text{et} \quad \lim_{\nu \rightarrow \infty} \min(0, -z_i^\nu \mid i \in I_+^\nu) = 0$$

$$(z^\nu = \Omega y^\nu)$$

alors $\{y^\nu\}$ est une suite minimisante".

Démonstration

D'après la démonstration de la proposition 2.4, comme la suite $\{F(y^\nu)\}$ est bornée, cela entraîne que la suite de P $\{y^\nu\}$ est bornée. On peut donc en extraire une suite $\{y^p\}$ qui converge vers $\bar{y} \in P$.

Posons $\bar{z} = \Omega \bar{y}$ et $\bar{I} = I(\bar{y})$.

On peut alors trouver un indice p_0 tel que :

$$(3.12) \quad p > p_0 \implies I^p \subset \bar{I}.$$

(a) soit $i \notin \bar{I}$. Comme $\bar{z}_i = \lim_{p \rightarrow \infty} z_i^p$, la propriété (3.11) entraîne :

$$\lim_{p \rightarrow \infty} |z_i^p| = 0 \quad \text{donc} \quad \bar{z}_i = 0.$$

(b) soit $i \in \bar{I}_-$

$$\lim_{p \rightarrow \infty} \max(0, -z_i^p) = 0 \quad \text{donc} \quad -\bar{z}_i \leq 0.$$

(c) soit $i \in \bar{I}_+$, alors

$$\lim_{p \rightarrow \infty} \min(0, -z_i^p) = 0 \text{ donc } z_i \leq 0$$

Le théorème de caractérisation 3.1 entraîne alors que \bar{y} est solution de (P1').

$$\text{Donc } d = \lim_{p \rightarrow \infty} F(y^p) = \lim_{\nu \rightarrow \infty} F(y^\nu),$$

puisque $\{y^\nu\}$ vérifie la propriété (3.10), et la suite $\{y^\nu\}$ est une suite minimisante.

Proposition 3.3

"L'algorithme n°1 converge vers la solution du problème (P1)"

Démonstration

Soit $y^0 \in P$ et $\{y^\nu\}$ la suite construite au moyen de l'algorithme décrit dans le paragraphe précédent. Nous supposons cette suite infinie.

Considérons la ν ième opération. Deux éventualités peuvent alors se présenter :

A. s^ν est la direction de descente "normale" au point y^ν .

Si $\lambda(y^\nu, s^\nu) = \lambda''(y^\nu, s^\nu)$ on a alors :

$$F(y^{\nu+1}) - F(y^\nu) \leq -\frac{1}{\gamma} \|s^\nu\|^2 < 0$$

Sinon on a $\lambda(y^\nu, s^\nu) = \lambda'(y^\nu, s^\nu)$ et $\|y^{\nu+1} - y^\nu\| \geq \delta$, ce qui entraîne :

$$F(y^{\nu+1}) - F(y^\nu) \leq \lambda - \delta \|s^\nu\| < 0$$

puisque $\|y^{\nu+1} - y^\nu\| = \lambda(y^\nu, s^\nu) \|s^\nu\|$.

B. s^ν est la direction de descente "modifiée".

Il est alors aisé de vérifier qu'après $n+1$ itérations au plus on a retrouvé le cas A ou que, sinon, on a obtenu la solution y_σ .

Finalement, comme la suite $\{\|s^\nu\|\}$ est bornée, on peut extraire de la suite $\{y^\nu\}$ une sous-suite infinie $\{y^{\nu_p}\}$ vérifiant :

$$(3.13) \quad F(y^{\nu_{p+1}}) - F(y^{\nu_p}) \leq M \|s^{\nu_p}\| < 0$$

et telle que, pour tout p , la direction s^{ν_p} soit la direction de descente "normale" au point s^{ν_p} .

Par construction, la suite $\{F(y^{\nu_p})\}$ est une suite décroissante positive. Donc elle converge et ceci entraîne :

$$\lim_{p \rightarrow \infty} \|s^{\nu_p}\| = 0$$

On déduit alors de la propriété (3.4) et du lemme 3.1 que la suite $\{y^{\nu_p}\}$ est une suite minimisante. Alors $\{F(y^{\nu_p})\}$ converge vers d et $\{y^{\nu_p}\}$ est une suite minimisante.

Il serait aisé de vérifier que dans le cas où $J \neq \emptyset$ nous pourrions établir un lemme analogue au lemme 3.1 dans lequel interviendrait J et que nous pourrions en déduire la convergence de l'algorithme n°1 quand $J \neq \emptyset$.

IV-ASPECTS NUMERIQUES ET PROCEDURES ALGOL-

4.1. Calcul de $\lambda(y^\nu, s^\nu)$

Calcul de $\lambda'(y^\nu, s^\nu) = \lambda'$

Soit $i \notin I^\nu$.

Si $s_i^\nu > 0$, on pose $d_i = \frac{b_i - y_i^\nu}{s_i^\nu}$

Si $s_i^\nu < 0$, on pose $d_i = \frac{a_i - y_i^\nu}{s_i^\nu}$

On a alors $\lambda' = \lambda'(y^\nu, s^\nu) = \min (d_i \mid i \notin I^\nu)$

Détermination de $\lambda(y^\nu, s^\nu)$

Soit $\bar{y} = y^\nu + \lambda' s^\nu$.

Alors $f'(\lambda') = (s^\nu \mid \bar{\Omega}_{\bar{y}})$.

Si $(s^\nu \mid \bar{\Omega}_{\bar{y}}) < 0$ on prend $\lambda(y^\nu, s^\nu) = \lambda'$ et $y^{\nu+1} = \bar{y}$.

Si $(s^\nu \mid \bar{\Omega}_{\bar{y}}) \geq 0$, on a $\lambda(y^\nu, s^\nu) = \lambda''(y^\nu, s^\nu)$

et $y^{\nu+1} = y^\nu + \lambda' \cdot \frac{(s^\nu \mid \Omega_{s^\nu})}{(s^\nu \mid \bar{\Omega}_{\bar{y}}) + (1+\lambda')(s^\nu \mid \Omega_{s^\nu})}$

4.2. Notice d'emploi de la procédure GRADPROJ1.

GRADPROJ1(N, Y, A, B, GRAD, PETI, DELTA, EPS, EPS1).

Cette procédure résout le problème (P1') dans le cas où P est borné et d'intérieur non vide.

Etant donné l'entier N, les tableaux A [1 : N] et B[1 : N] définissent le pavé P de \mathbb{R}^N :

$$P = \{X \in \mathbb{R}^N \mid A[I] \leq X[I] \leq B[I], I = 1, \dots, N\}.$$

Etant donnée la fonction $F(y) = \sum_{i=1}^N y_i (\alpha y)_i$, la procédure permet d'obtenir de

façon itérative, à partir d'un vecteur y^0 le vecteur y_σ "minimisant" F sur P .

A l'appel de la procédure, le tableau $Y[1 : N]$ doit contenir y^0 . A chaque itération ν , le tableau Y contient y^ν et le tableau $MU[1 : N]$ contient le gradient de F en y^ν . MU est calculé par la procédure $GRAD(N, Y, MU)$.

Le réel $DELTA$ détermine la modification éventuelle de la direction de descente.

La procédure comporte deux tests d'arrêt :

1) à chaque itération, l'algorithme calcule $\|s^\nu\|^2$. On arrête les calculs si :

$$\|s^\nu\|^2 < EPS1.$$

2) le calcul s'arrête dès que $\left| \frac{F(y^\nu) - F(y^{\nu+1})}{F(y^\nu)} \right| < EPS$.

4.3. Notice d'emploi de la procédure GRADPROJ2.

$GRADPROJ2(N, Y, A, SG, VAR, GRAO, PETI, DELTA, EPS, EPS1)$

Cette procédure a la même structure que la procédure $GRADPROJ1$. Elle résoud le problème (P1') et s'utilise, de préférence à $GRADPROJ$, dans le cas où le nombre des indices i , pour lesquels α_i ou β_i a une valeur infinie, est élevé.

Si $I \in J$ ($1 \leq I \leq N$) alors $VAR[I] = 1$, sinon $VAR[I] = 0$.

L'entier M donne le nombre de contraintes qui détermine le pavé $P(M \geq N)$. Etant donné le tableau $A[1 : M]$ et l'entier tableau $SG[1 : M]$, on a :

$$P = \{X \in \mathbb{R}^N \mid \text{SIGN}(SG[I]) \times X(\text{ABS}(SG[I])) \leq A[I] \quad I = 1, \dots, M\}.$$


```

PROCEDURE GRADPROJ1(N,Y,A,B,GRAD,PETI,DELTA,EPS,EPS1);
INTEGER N;
ARRAY Y,A,B;
PROCEDURE GRAD;
REAL PETI,DELTA,EPS,EPS1;

BEGIN
  INTEGER I,J;
  REAL D,DP,E,F,FCT0,FCT1;
  REAL ARRAY S,Y1,MU,MU1(/I:N/);
  INTEGER ARRAY FRONT(/I:N/);
  FCT1:=0;
  ITER1:GRAD(N,Y,MU);
  ITER2:FCT0:=0;
  FOR I:=1 STEP 1 UNTIL N DO FCT0:=FCT0+Y(/I/)*MU(/I/);
  IF ABS((FCT0-FCT1)/FCT0)<EPS THEN GO TO FINAL;
  FOR I:=1 STEP 1 UNTIL N DO
    BEGIN
      IF (Y(/I/)-A(/I/))<PETI THEN FRONT(/I/):=I ELSE
      IF (B(/I/)-Y(/I/))<PETI THEN FRONT(/I/):=-I
      ELSE FRONT(/I/):=0
    END ;
    DIRECTION:
    FOR I:=1 STEP 1 UNTIL N DO
      BEGIN
        IF FRONT(/I/)*MU(/I/)<=0 THEN S(/I/):=-MU(/I/)
        ELSE S(/I/):=0
      END ;
      D:=0;
      FOR I:=1 STEP 1 UNTIL N DO D:=D+S(/I/)*S(/I/);
      IF D<EPS1 THEN GO TO FINAL;
      CHOIX DU PAS:
      DP:=0:0;
      FOR I:=1 STEP 1 UNTIL N DO
        BEGIN
          IF S(/I/)>0 THEN D:=(B(/I/)-Y(/I/))/S(/I/) ELSE
          IF S(/I/)<0 THEN D:=(A(/I/)-Y(/I/))/S(/I/);
          IF (DP=0) OR ((D<DP) AND (DP/=0)) THEN
            BEGIN
              DP:=D;
              J:=I
            END ;
          END ;
          FOR I:=1 STEP 1 UNTIL N DO Y1(/I/):=Y(/I/)+DP*S(/I/);
          IF S(/J/)>0 THEN Y1(/J/):=B(/J/) ELSE IF S(/J/)<0
          THEN Y1(/J/):=A(/J/);
          GRAD(N,Y1,MU1);
          E:=F:=0:0;
          FOR I:=1 STEP 1 UNTIL N DO
            BEGIN

```



```

E:=S(/I/)*MU1(/I/)+E;
F:=S(/I/)*S(/I/)+F;
END ;
IF E>0 THEN
BEGIN
DP:=DP*F/(F+E);
FOR I:=1 STEP 1 UNTIL N DO Y(/I/):=Y(/I/)+DP*S(/I/);
FCT1:=FCT0;
GO TO ITER1
END
ELSE IF E<0 THEN
BEGIN
D:=SQRT(F)*DP;
IF D<DELTA THEN GO TO ANTIZIG;
FOR I:=1 STEP 1 UNTIL N DO
BEGIN
Y(/I/):=Y1(/I/);
MU(/I/):=MU1(/I/);
END ;
FCT0:=FCT1;
GO TO ITER2
END ;
ANTIZIG: FOR I:=1 STEP 1 UNTIL N DO Y1(/I/):=Y(/I/);
IF S(/J/)>0 THEN
BEGIN
Y1(/J/):=B(/J/);
DP:=(B(/J/)-Y(/J/))/S(/J/)
END
ELSE
BEGIN
Y1(/J/):=A(/J/);
DP:=(A(/J/)-Y(/J/))/S(/J/)
END ;
GRAD(N,Y1,MU1);
E:=S(/J/)*MU1(/J/);
IF E>0 THEN
BEGIN
F:=S(/J/)*S(/J/);
DP:=DP*F/(F+E);
Y(/J/):=Y(/J/)+DP*S(/J/);
GRAD(N,Y,MU);
END
ELSE IF E<0 THEN
BEGIN
FOR I:=1 STEP 1 UNTIL N DO MU(/I/):=MU1(/I/);
Y(/J/):=Y1(/J/);
END ;
FCT1:=FCT0;
GO TO ITER2;
FINAL:
END GRADPROJ1;

```



```

PROCEDURE GRADPROJ2(N,M,X,A,SG,VAR,Y,EPS,EPS1,DELTA,PETI,GRAD);
INTEGER N,M;
ARRAY X,Y,A;
INTEGER ARRAY SG,VAR;
REAL EPS,DELTA,PETI,EPS1;
PROCEDURE GRAD;

```

```

BEGIN

```

```

    INTEGER I,I1,J;
    REAL F,F1,D,DP,E,E1;
    INTEGER ARRAY FR(/1:N/);
    ARRAY S,Y1,MU,MU1(/1:N/);
    F:=0;
    ITER1: GRAD(N,Y,MU);
    ITER3: FOR I:=1 STEP 1 UNTIL N DO FR(/I/):=0;
    ITER2: F1:=0;
    FOR I:=1 STEP 1 UNTIL N DO F1:=F1+Y(/I/)*MU(/I/);
    IF ABS((F1-F)/F1)<EPS THEN GOTO FINAL;
    FOR I:=1 STEP 1 UNTIL M DO
        BEGIN
            I1:=SG(/I/);
            IF ABS(A(/I/)-SIGN(I1)*Y(/ABS(I1)/))<PETI THEN FR(/ABS(I1)/):=SIGN(I1)
        END ;
    DIRECTION: FOR I:=1 STEP 1 UNTIL N DO IF FR(/I/)*MU(/I/)>=0 & VAR(/I/)=0 THEN S(/I/):=-MU(/I/) ELSE S(/I/):=0;
    E1:=0;
    FOR I:=1 STEP 1 UNTIL N DO E1:=S(/I/)*S(/I/)+E1;
    IF E1<EPS1 THEN GOTO FINAL;
    CHOIXDUPAS: DP:=0;
    FOR I:=1 STEP 1 UNTIL M DO
        BEGIN
            I1:=SG(/I/);
            IF SIGN(I1)*S(/ABS(I1)/)>*-8 THEN
                BEGIN
                    D:=(A(/I/)-SIGN(I1)*Y(/ABS(I1)/))/ABS(S(/ABS(I1)/));
                    IF (DP=0) | ((D<DP) & (DP<=0)) THEN
                        BEGIN
                            DP:=D;
                            J:=I
                        END
                    END
                END
        END ;
    Y1(/ABS(SG(/J/))):=A(/J/);
    FOR I:=1 STEP 1 UNTIL N DO Y1(/I/):=Y(/I/)+DP*S(/I/);
    GRAD(N,Y1,MU1);
    E:=0;
    FOR I:=1 STEP 1 UNTIL N DO E:=S(/I/)*MU1(/I/)+E;
    IF E>0 THEN
        BEGIN

```

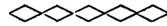


```

DP:=DP*E1/(E1+E);
FOR I:=1 STEP 1 UNTIL N DO Y(/I/):=Y(/I/)+DP*S(/I/);
F:=F1;
GOTO ITER1
END ;
IF E<0 THEN
BEGIN
D:=DP*SQRT(E1);
IF D<DELTA THEN GOTO ANTIZIG;
FOR I:=1 STEP 1 UNTIL N DO
BEGIN
Y(/I/):=Y1(/I/);
MU(/I/):=MU1(/I/)
END ;
F:=F1;
GOTO ITER3
END ;
ANTIZIG: FOR I:=1 STEP 1 UNTIL N DO Y1(/I/):=Y(/I/);
I1:=SG(/J/);
Y1(/ABS(I1)/):=A(/J/);
D:=(A(/J/)-SIGN(I1)*Y(/ABS(I1)/))/ABS(S(/ABS(I1)/));
GRAD(N,Y1,MU1);
E:=S(/ABS(I1)/)*MU1(/ABS(I1)/);
E1:=S(/ABS(I1)/)*S(/ABS(I1)/);
IF E>0 THEN
BEGIN
DP:=DP*E1/(E+E1);
Y(/ABS(I1)/):=Y(/ABS(I1)/)+DP*S(/ABS(I1)/)
END
ELSE IF E<0 THEN Y(/ABS(I1)/):=A(/J/);
GRAD(N,Y,MU);
F:=F1;
GOTO ITER2;
FINAL:
END GRADPROJ2;

```


C H A P I T R E I V



METHODES DE RELAXATION

Les méthodes exposées dans ce chapitre permettent de résoudre le problème (P1') défini au chapitre III. Elles s'inspirent des méthodes de relaxation utilisées pour résoudre des systèmes linéaires (cf Gastinel [11]) : le principe de ces méthodes est rappelé dans un premier paragraphe. Les deux paragraphes suivants sont consacrés à la description de deux méthodes et à l'étude de leurs convergences. Dans un quatrième paragraphe nous proposons des procédures Algol.

I -PRINCIPES DES METHODES DE RELAXATION-

Rappels

Trouver la solution du système :

$$Ax = b \quad , \quad x \text{ et } b \in \mathbb{R}^n ,$$

où A est une matrice symétrique définie positive, c'est trouver le point \bar{x} où la fonction

$$E(x) = \frac{1}{2} x^T Ax - x^T b$$

atteint son minimum.

$Ax-b$ est le gradient de E en x.

Les méthodes de relaxation consistent à construire des suites $\{x^\nu\}$ de \mathbb{R}^n qui convergent vers \bar{x} et qui sont telles, que pour tout ν , x^ν et $x^{\nu+1}$ ne diffèrent que par une composante.

Dans la méthode de Southwell, étant donné x^ν , on choisit j^ν vérifiant :

$$|(Ax^\nu - b)_{j^\nu}| = \max(|(Ax^\nu - b)_i|, i \in \{1, \dots, n\}),$$

et on pose $x_i^{\nu+1} = x_i^\nu$ pour tout $i \neq j^\nu$. $x_{j^\nu}^{\nu+1}$ est alors choisi de façon à ce que $E(x^{\nu+1})$ soit le plus petit possible.

Dans la méthode de Gauss-Seidel on s'impose j^ν : étant donné x^ν et l'indice $i^\nu \in \{1, \dots, n\}$, on pose :

$$j^\nu = i^\nu + 1 \pmod{n} = i^{\nu+1},$$

et $x^{\nu+1}$ est déterminé comme dans méthode de Southwell.

Le problème (P1') consiste à trouver le minimum de la forme quadratique dégénérée :

$$F(y) = y^T \Omega y$$

sur le polyèdre convexe P de \mathbb{R}^n :

$$P = \{y \in \mathbb{R}^n \mid \alpha_i \leq y_i \leq \beta_i \quad i = 1, \dots, n\}$$

Etant donné $x^\nu \in P$ nous choisirons $j^\nu \in \{1, \dots, n\}$ (ce choix sera précisé ultérieurement) et nous poserons :

$$x_i^{\nu+1} = x_i^\nu \quad \text{pour tout } i \neq j^\nu.$$

Alors $x_{j^\nu}^{\nu+1}$ sera choisi de telle sorte que $x^{\nu+1}$ soit dans P et que $F(x^{\nu+1})$ ait la plus petite valeur possible.

II-ALGORITHME N°2 (cf Méthode de SOUTHWELL).

2.1. Itération.

Soit $y^\nu \in P$. On suppose que y^ν n'est pas solution. On définit z^ν, I_+^ν et I_-^ν comme au paragraphe 2.1 du chapitre III.

a) Choix de l'indice j^v

Soit t^v le vecteur défini par :

$$t_i^v = -z_i^v \quad \text{si } i \notin I^v$$

$$t_i^v = \min(-z_i^v, 0) \quad \text{si } i \in I_+^v - J$$

$$t_i^v = \max(-z_i^v, 0) \quad \text{si } i \in I_-^v - J$$

$$t_i^v = 0 \quad \text{si } i \in J.$$

Remarque :

t^v est le vecteur s^v défini dans l'algorithme n°1.

Puisque y^v n'est pas solution, t^v n'est pas nul et on choisit un indice j^v vérifiant :

$$(4-1) \quad |t_{j^v}^v| = \max(|t_i^v|, i \in \{1, \dots, n\}) > 0$$

La direction de descente sera la direction s^v avec :

$$(4-2) \quad s_{j^v}^v = t_{j^v}^v \quad \text{et} \quad s_i^v = 0 \quad \text{pour tout } i \neq j^v.$$

b) Choix du pas

Le pas λ^v sera en core défini par :

$$\lambda^v = \min(\lambda'(s^v, y^v), \lambda''(s^v, y^v)).$$

Remarque :

On peut vérifier que $s^v \in \Gamma^*(P, y^v) \cap \Gamma(Q, y^v)$

(On a encore : $\|s^v\|^2 = -(s^v | \Omega y^v)$).

2.2 Convergence de l'algorithme :

Proposition 4.1.

"L'algorithme n°2 converge vers la solution du problème (P1)".

Démonstration (cf démonstration de la convergence de l'algorithme n°1)

Soit $y^0 \in P$. On suppose que la suite $\{y^\nu\}$ définie par l'algorithme n°2 est infinie. A la ν ème itération, trois éventualités sont possibles :

$$\textcircled{1} \lambda^\nu = \lambda''(y^\nu, s^\nu).$$

On a alors : $F(y^{\nu+1}) - F(y^\nu) \leq -\frac{1}{\gamma} \|s^\nu\|^2 < 0$

$$\textcircled{2} \lambda^\nu = \lambda'(y^\nu, s^\nu) \text{ avec } \|y^{\nu+1} - y^\nu\| = \left| y_{j^\nu}^{\nu+1} - y_{j^\nu}^\nu \right| \geq \delta$$

où δ est un nombre positif donné, "assez petit".

Alors : $F(y^{\nu+1}) - F(y^\nu) \leq -\delta \|s^\nu\| < 0$.

$$\textcircled{3} \lambda^\nu = \lambda'(y^\nu, s^\nu) \text{ avec } \|y^{\nu+1} - y^\nu\| < \delta$$

On peut vérifier alors qu'ayant $n+1$ itérations au plus, soit le cas $\textcircled{1}$ aura été vérifié, soit on sera sur un sommet et à l'itération suivante le cas $\textcircled{1}$ ou le cas $\textcircled{2}$ sera vérifié (Il suffit que δ soit inférieur à $\min(\beta_i - \alpha_i \mid i \notin J)$).

Comme la suite $\{y^\nu\}$ est bornée, on peut extraire de la suite $\{y^\nu\}$ une sous-suite infinie $\{y^{\nu(p)}\}$ vérifiant :

$$(4.3) F(y^{\nu(p)+1}) - F(y^{\nu(p)}) \leq M \|s^{\nu(p)}\| < 0$$

où M est une constante positive.

On peut alors conclure, par le même raisonnement que celui qui a été développé dans la démonstration de la convergence de l'algorithme n°1, que la suite $\{y^\nu\}$ est une suite minimisante.

III-ALGORITHME N°3 (cf Méthode de GAUSS-3.1. Itération.

Soit $y^\nu \in P$. On définit alors une suite finie $\{x^{\nu,i}\}_{i=1,\dots,n}$ d'éléments de P au moyen de la procédure suivante :

$$(a) \quad x^\nu = y^\nu$$

$$(b) \quad i \in \{1, \dots, n\}$$

Soit $s_i \in \mathbb{R}^n$, $s_i^j = \begin{cases} 0 & \text{si } j \neq i \\ 1 & \text{si } j = i \end{cases}$ et λ^i défini par :

$$(4.5) \quad \lambda^i = \lambda(x^{\nu,i}, s_i) = \min(\lambda'(x^{\nu,i}, s_i) = \min(\lambda'(x^{\nu,i}, s_i), \lambda''(x^{\nu,i}, s_i))).$$

On pose $x^{\nu,i+1} = x^{\nu,i} + \lambda^i s_i$

$$(c) \quad y^{\nu+1} = x^{\nu,n+1} = x^{\nu+1,n}$$

Remarques.

Si y^ν est solution de $(P1')$, alors $\lambda^i = 0$ pour tout i et on obtient $y^{\nu+1} = y^\nu$.

Si y^ν n'est pas solution, alors $F(y^{\nu+1}) < F(y^\nu)$.

On peut aussi remarquer que pour tout i tel que $\lambda^i \neq 0$, alors

$$(4.6) \quad s_i \in \Gamma^*(P, x^{\nu,i}) \cap \Gamma(Q_x^{\nu,i}, x^{\nu,i})$$

$$= \Gamma^*(P, y^\nu) \cap \Gamma(Q_y^{\nu,i}, y^\nu).$$

3.2. Propriétés de la suite $\{x^{\nu,i}\}$ ($i = 1, \dots, n$).

A chaque itération ν et pour chaque indice i considérons :

$$(4.7) \quad z^{\nu,i} = \Omega x^{\nu,i} \in \mathbb{R}^n$$

et $t^{i,\nu} \in \mathbb{R}$ défini par (cf propriété (3.4)) :

$$\begin{cases} t^{\nu,i} = -z_i^{\nu,i} & \text{si } i \notin I^\nu = I(y^\nu) \\ t^{\nu,i} = \min(-z_i^{\nu,i}, 0) & \text{si } i \in I_+^\nu - J \\ t^{\nu,i} = \max(-z_i^{\nu,i}, 0) & \text{si } i \in I^\nu - J \\ t^{\nu,i} = 0 & \text{si } i \in J \end{cases}$$

Quatre cas sont alors possibles :

$$\textcircled{a} \quad \lambda^i = 0 \iff t^{\nu,i} = 0$$

$$\textcircled{b} \quad \lambda^i = \lambda''(x^{\nu,i}, s_i)$$

$$\text{alors on a : } F(x^{\nu,i+1}) - F(x^{\nu,i}) \leq -\frac{1}{\gamma} |t^{\nu,i}|^2 < 0.$$

$$\textcircled{c} \quad \lambda^i = \lambda'(x^{\nu,i}, s_i) \neq 0 \quad \text{et} \quad |x_i^{\nu,i+1} - x_i^{\nu,i}| \geq \delta$$

$$\text{alors : } F(x^{\nu,i+1}) - F(x^{\nu,i}) \leq -\delta |t^{\nu,i}| < 0$$

\textcircled{d} Si les cas précédents ne sont pas vérifiés, on voit qu'à l'itération $\nu+1$ ième et pour l'indice i , l'un de ces trois cas sera vérifié (il suffit de choisir δ inférieur à $\min(\beta_i - \alpha_i \mid i \notin J)$).

3.3. Convergence de l'algorithme n°3

Proposition 4.2.

"L'algorithme n°3 converge vers la solution du problème (P1)".

Démonstration

Soit $y^0 \in P$ et $\{y^\nu\}$ la suite définie par l'algorithme n°3. On suppose

cette suite infinie. Comme $\{F(y^\nu)\}$ est alors une suite strictement décroissante, la suite $\{y^\nu\}$ est une suite bornée et on peut en extraire une sous-suite convergente $\{y^{\nu(p)}\}$ ayant la propriété suivante :

$$(4.8) \text{ si } \lim_{p \rightarrow \infty} y^{\nu(p)} = \bar{y} \text{ alors } \{y^{\nu(p)+1}\} \text{ converge et } \lim_{p \rightarrow \infty} y^{p(\nu)+1} = \bar{y}.$$

Nous allons montrer que $\{y^{\nu(p)}\}$ est une suite minimisante; comme la suite $\{F(y^\nu)\}$ converge, on en déduira que $\{y^\nu\}$ est une suite minimisante, d'où la proposition.

Considérons pour un indice i fixé, $i \in \{1, \dots, n\}$, la suite $\{t^{\nu(p), i}\}$. D'après la propriété (4.8) et les propriétés étudiées au paragraphe précédent, on sait qu'on peut extraire de cette suite une sous-suite $\{y^{\nu(q), i}\}$ qui converge vers 0. D'autre part, la convergence de $\{y^{\nu(p)}\}$ vers \bar{y} entraîne la convergence de $\{z^{\nu(p)}\}$, où $z^{\nu(p)} = \Omega y^{\nu(p)}$, vers $\bar{z} = \Omega \bar{y}$. On en déduit alors

$$(4.9) \lim_{p \rightarrow \infty} t^{\nu(p), i} = \lim_{q \rightarrow \infty} t^{\nu(q), i} = 0$$

On pourrait alors montrer (cf démonstration de la convergence de l'algorithme n°1, que la propriété (4.9) entraîne que \bar{y} vérifie la propriété (3.3) donc que \bar{y} est solution du problème (P1') la suite $\{y^{\nu(p)}\}$ est donc bien une suite minimisante.

IV-PROCEDURES ALGOL-

Notice d'emploi des procédures SOUTHWELL et GAUSS-SEIDEL-

SOUTHWELL (N, A, B, Y, S, EPS1, EPS2).

GAUSS SEIDEL(N, A, B, Y, S, EPS1, EPS2).

Ces procédures résolvent le problème (P1') au moyen, respectivement, de l'algorithme n°2 et de l'algorithme n°3.

Etant donné l'entier N, les tableaux A[1 : N] et B[1 : N] définissent le pavé P de \mathbb{R}^N :

$$P = \{X \in \mathbb{R}^N \mid A[I] \leq X[I] \leq B[I] \quad I = 1 \dots N\}$$

Etant donnée la fonction $F(y) = \sum_{i=1}^N \sum_{j=1}^N y_i y_j \Omega_i^j$, les procédures permettent d'obtenir de façon itérative, à partir d'un vecteur y^0 , le vecteur y_0 pour lequel F atteint son minimum sur P. La fonction est précisée par la donnée du tableau S[1 : N, 1 : N] qui contient les éléments de la matrice Ω_i^j .

GAUSS-SEIDEL : à l'appel de la procédure, le tableau Y[1 : N] doit contenir y^0 . A chaque itération (ν, i), le tableau Y contient $x''^{i, \nu}$. La procédure comporte deux tests d'arrêt :

1) TEST1 : il vérifie après l'itération (ν, N) si le point $y^{\nu+1}$ est solution ou non (avec la précision EPS1 sur le gradient).

2) TEST2 : On arrête les calculs si $\frac{|F(y'') - F(y^{\nu+1})|}{F(y'')} < \text{EPS2}$.

SOUTHWELL : à l'appel de la procédure, le tableau Y[1 : N] doit contenir y'' . A chaque itération ν , le tableau Y contient y'' . La procédure comporte deux tests-d'arrêt :

1) TEST1 : à chaque itération, on vérifie si on a la solution ou non, avec la précision EPS1 sur le gradient.

2) TEST2 : on arrête les calculs dès que $\frac{|F(y''^{\nu}) - F(y''^{\nu+N})|}{F(y''^{\nu})} < \text{EPS2}$.

```

PROCEDURE GAUSSEI(N,A,B,Y,G,S,EPS1,EPS2);
INTEGER N;
ARRAY A,B,Y,S,G;
REAL EPS1,EPS2;

```

```

BEGIN

```

```

  INTEGER I,J;
  INTEGER ARRAY FRONT(/1:N/);
  REAL F,F1,R,R1;
  F:=0;
  FOR I:=1 STEP 1 UNTIL N DO

```

```

    BEGIN

```

```

      FOR J:=1 STEP 1 UNTIL N DO G(/I/):=G(/I/)+Y(/J/)*S(/J,I/);
      F:=F+G(/I/)*Y(/I/);
      FRONT(/I/):=0

```

```

    END ;

```

```

  I:=0;

```

```

  ITER:I:=I+1;

```

```

  IF G(/I/)=0 THEN GOIQ SUI2;

```

```

  R:=-G(/I/)/S(/I,I/);

```

```

  IF G(/I/)<0 THEN

```

```

    BEGIN

```

```

      IF Y(/I/)+R<B(/I/) THEN Y(/I/):=Y(/I/)+R ELSE

```

```


```

```

      R:=B(/I/)-Y(/I/);

```

```

      FRONT(/I/):=1;

```

```

      Y(/I/):=B(/I/)

```

```

    END ;

```

```

  END

```

```

  ELSE IF G(/I/)>0 THEN

```

```

    BEGIN

```

```

      IF Y(/I/)+R>A(/I/) THEN Y(/I/):=Y(/I/)+R ELSE

```

```


```

```

      R:=A(/I/)-Y(/I/);

```

```

      FRONT(/I/):=-1;

```

```

      Y(/I/):=A(/I/)

```

```

    END ;

```

```

  END ;

```

```

  SUI1: FOR J:=1 STEP 1 UNTIL N DO G(/J/):=G(/J/)+R*S(/I,J/);

```

```

  SUI2: IF I<N THEN GOIQ ITER;

```

```

  R:=0;

```

```

  FOR J:=1 STEP 1 UNTIL N DO

```

```

    BEGIN

```

```

      IF FRONT(/J/)=0 THEN R1:=ABS(G(/J/)) ELSE R1:=G(/J/)*FRONT(/J/);

```

```

      IF R1>R THEN R:=R1

```

```

    END ;

```

```

  IF R<EPS1 THEN GOIQ FINAL;

```

```

  F1:=0;

```

```

  FOR J:=1 STEP 1 UNTIL N DO F1:=F1+G(/J/)*Y(/J/);

```

```

  TEST2: IF ABS((F1-F)/F)<EPS2 THEN GOIQ FINAL;

```

```

  F:=F1;

```

```

  I:=0;

```

```

  GOIQ ITER;

```

```

  FINAL:

```

```

END GAUSS SEIDEL;

```



```

PROGRAM SOUTH(N,A,B,Y,G,S,EPS1,EPS2);
INTEGER N;
ARRAY A,B,Y,S,G;
REAL EPS1,EPS2;

```

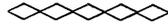
```

BEGIN
  INTEGER ARRAY FRONT(1:N);
  REAL F,F1,R,R1;
  INTEGER I,J,L1,L;
  F:=0;
  L:=L1:=0;
  FOR I:=1 STEP 1 UNTIL N DO
    BEGIN
      FOR J:=1 STEP 1 UNTIL N DO G(I/J):=G(I/J)+Y(J/I)*S(J,I);
      F:=F+G(I/I)*Y(I/I);
      FRONT(I/I):=0
    END ;
    R:=R:=0;
    I:=1;
    L1:=L1+1;
    FOR J:=1 STEP 1 UNTIL N DO
      BEGIN
        IF FRONT(J/I)=0 THEN R1:=ABS(G(J/I)) ELSE R1:=G(J/I)*FRONT(J/I);
        IF R1>R THEN
          BEGIN
            R:=R1;
            I:=J
          END ;
        END ;
      END ;
      IF (I=0) OR (R<EPS1) THEN GOIC BUT;
      R:=-G(I/I)/S(I,I);
      IF G(I/I)<0 THEN
        BEGIN
          IF Y(I/I)+R<B(I/I) THEN Y(I/I):=Y(I/I)+R ELSE
            BEGIN
              R:=B(I/I)-Y(I/I);
              FRONT(I/I):=1;
              Y(I/I):=B(I/I)
            END ;
        ELSE IF G(I/I)>0 THEN
          BEGIN
            IF Y(I/I)+R>A(I/I) THEN Y(I/I):=Y(I/I)+R ELSE
              BEGIN
                R:=A(I/I)-Y(I/I);
                FRONT(I/I):=-1;
                Y(I/I):=A(I/I)
              END ;
            END ;
          END ;
          R:=R;
          FOR J:=1 STEP 1 UNTIL N DO G(J/I):=G(J/I)+R*S(I,J/I);

```



```
IF L1=N THEN
  BEGIN
    F1:=0;
    FOR J:=1 STEP 1 UNTIL N DO F1:=F1+G(/J/)*Y(/J/);
    IF ABS((F1-F)/F)<EPS2 THEN GOIC BUT;
    F:=F1;
    L:=L+1;
    L1:=0
  END ;
  GOIC RTS;
  BUT:
END SOUTHWELL;
```

" METHODES DE FRANCK ET WOLFE "

Dans ce chapitre nous présentons deux algorithmes permettant de résoudre le problème (P1) défini au chapitre II. Ces algorithmes sont une généralisation de l'algorithme de Franck et Wolfe (cf Gilbert [12], Haugazeau [14], Céa[8]).

Dans un premier paragraphe, nous décrivons ces algorithmes. Nous en étudions la convergence au paragraphe 2. Le troisième paragraphe est consacré à des procédures Algol.

I-DESCRIPTION DES ALGORITHMES N°4 ET N°4'-(cf Céa [8])

1.1. Soit $y \in P$ et K un voisinage convexe et fermé de y . Soit $H(y)$ l'hyperplan pour $K \cap P$ défini par :

$$H(y) = \{x \in \mathbb{R}^n \mid (x \mid \Omega y) = \min [(z \mid \Omega y) \mid z \in P \cap K]\}.$$

Il est aisé de vérifier (cf proposition 2.2) qu'une condition nécessaire et suffisante pour que $\bar{y} \in P$ minimise F sur le convexe $P \cap K$ est que $\bar{y} \in H(\bar{y})$. F étant convexe, tout minimum "local" sur P est un minimum "global". On en déduit le résultat suivant :

Lemme 5.1. "Une condition nécessaire et suffisante pour que \bar{y} soit solution du problème (P1) est que $\bar{y} \in H(\bar{y})$ ".

1.2. Description de l'algorithme N°4.

Soit $y^\nu \in P$ et K^ν un voisinage convexe et fermé de y^ν dont le diamètre est compris entre r et R ($0 < r \leq R$). Supposons que y^ν ne soit pas solution.

Alors $y^\nu \notin H(y^\nu)$ d'après le lemme 5.1 et on peut trouver x^ν , $x^\nu \neq y^\nu$, tel que $x^\nu \in H(y^\nu) \cap P \cap K^\nu$. On prend la direction $x^\nu - y^\nu$ comme direction de descente et on pose :

$$y^{\nu+1} = y^\nu + \lambda^\nu (x^\nu - y^\nu)$$

avec $\lambda^\nu = \min(\lambda^1(y^\nu, x^\nu - y^\nu), \lambda^2(y^\nu, x^\nu - y^\nu))$.

Remarques.

- 1) Si $H(y^\nu) \cap P \cap K^\nu$ n'est pas réduit à un point, il y a une infinité de choix possibles pour x^ν .
- 2) x^ν et $y^\nu \in P$; alors $x^\nu - y^\nu \in \Gamma^*(P, y^\nu)$. D'autre part $y^\nu \in K^\nu \cap P$ et $y^\nu \notin H(y^\nu)$. Ceci entraîne l'inégalité :

$$(x^\nu | \Omega y^\nu) < (y^\nu | \Omega y^\nu)$$

et, d'après le lemme 2.1, $x^\nu - y^\nu \in \Gamma(Q_{y^\nu}, y^\nu)$

Donc $x^\nu - y^\nu \in \Gamma^*(P, y^\nu) \cap \Gamma(Q_{y^\nu}, y^\nu)$.

- 3) si P est borné, on peut choisir K^ν de sorte que $K^\nu \cap P = P$. L'algorithme n°4 est alors l'algorithme de Franck et Wolfe (cf "Algorithme des segments successifs" Haugazeau [14]).

1.3 Propriétés des suites $\{x^\nu\}$ et $\{y^\nu\}$

Notation

Afin de mieux voir les propriétés géométriques de l'algorithme, nous posons :

$$((x | y)) = (x | \Omega y) \quad \forall x, y \in \mathbb{R}^n.$$

L'application qui, au couple, $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$, fait correspondre le nombre $((x | y))$ est un semi produit scalaire.

(5.1) la suite $\{F(y^\nu)\}$ est une suite décroissante.

Comme $y^{\nu+1}$ est le minimum de F sur un sous-ensemble de P contenant y^ν et x^ν , on a les inégalités :

$$(5.2) \quad ((y^{\nu+1} \mid y^\nu - y^{\nu+1})) \geq 0 \text{ et}$$

$$(5.3) \quad ((y^{\nu+1} \mid x^\nu - y^{\nu+1})) \geq 0.$$

Par définition de $H(y^\nu)$ on a :

$$(5.4) \quad ((x^\nu - z \mid y^\nu)) \leq 0 \quad \forall z \in P \cap K^\nu.$$

1.4 Description de l'algorithme n°4. (cf. Cea [8].

Cet algorithme est une variante de l'algorithme n°4; Il consiste à essayer de choisir à la ν ième itération une direction de descente meilleure que la direction que l'on aurait pu choisir avec l'algorithme n°4.

Soit $y^\nu = y^{\nu-1} + \lambda^{\nu-1} s^{\nu-1}$ où $s^{\nu-1}$ est la direction de descente choisie à la $(\nu-1)$ ième itération et

$$\lambda^{\nu-1} = \min (\lambda'(y^{\nu-1}, s^{\nu-1}), \lambda''(y^{\nu-1}, s^{\nu-1})) > 0.$$

On suppose que y^ν n'est pas solution. Soit $x^\nu \in H(y^\nu) \cap P \cap K^\nu$.

A la ν ième itération la direction de descente est fonction de y^ν et de la $(\nu-1)$ ième itération. Deux cas sont possibles :

$$\textcircled{a} \quad \lambda^{\nu-1} = \lambda'(y^{\nu-1}, s^{\nu-1}) < \lambda''(y^{\nu-1}, s^{\nu-1}).$$

On prend encore comme dans l'algorithme n°4, la direction de descente $s^\nu = x^\nu - y^\nu$.

$$\textcircled{b} \quad \lambda^{\nu-1} = \lambda''(y^{\nu-1}, s^{\nu-1})$$

On a alors : $((y^\nu \mid s^{\nu-1})) = 0$ ce qui entraîne pour tout réel ρ :

$$((x^\nu + \rho s^{\nu-1} \mid y^\nu)) = ((x^\nu \mid y^\nu)) \leq ((z \mid y^\nu)) \quad \forall z \in P \cap K^\nu.$$

et $x^\nu + \rho s^{\nu-1} \in H(y^\nu)$ pour tout $\rho \in \mathbb{R}$.

Soit $\rho \in \mathbb{R}$ et $\mu_\rho = \lambda''(y^v, x^v + \rho s^{v-1} - y^v)$.

Alors, si on pose $p_\rho = x^v + \rho s^{v-1} - y^v$, on a :

$$F(y^v + \mu_\rho p_\rho) = \min (F(y^v + \lambda p_\rho) \mid \lambda \in \mathbb{R})$$

et il est facile de vérifier (cf propriété (2-7)) :

$$F(y^v + \mu_\rho p_\rho) - F(y^v) = - \frac{((p_\rho \mid y^v))^2}{F(p_\rho)} = - \frac{((x^v - y^v \mid y^v))^2}{F(p_\rho)}$$

Nous allons étudier les variations de $F(y^v + \mu_\rho p_\rho)$ en fonction de ρ .

$$F(p_\rho) = F(x^v + \rho s^{v-1} - y^v) = F(x^v - y^v) + 2((s^{v-1} \mid x^v - y^v)) + \rho^2 F(s^{v-1})$$

d'après la propriété (2-7).

Comme $y^v \neq y^{v+1}$, $F(s^{v-1}) \neq 0$ et on peut poser :

$$\rho_m = - \frac{((s^{v-1} \mid x^v - y^v))}{F(s^{v-1})} = - \frac{((s^{v-1} \mid x^v))}{F(s^{v-1})}$$

$F(y^v + \mu_\rho p_\rho)$ est une fonction de ρ strictement décroissante sur $]-\infty, \rho_m]$ et strictement croissante sur $[\rho_m, \infty[$. Dans le plan passant par y^v et parallèle aux vecteurs s^{v-1} et $x^v - y^v$, $y^v + \mu_{\rho_m} p_{\rho_m}$ est le point où la fonction F atteint son minimum.

Nous poserons :

$$\rho^v = \max \left\{ \rho \in \mathbb{R} \mid x^v + \rho s^{v-1} \in P, |\rho| < \max(\tau, |\rho_m|), \text{signe}(\rho) = \text{signe}(\rho_m) \right\}$$

où τ est un nombre réel donné.

En prenant pour direction de descente $s^v = x^v - y^v + \rho^v s^{v-1}$ nous tentons donc d'avoir une décroissance "la plus rapide possible" pour $\{F(y^v)\}$.

Remarque :

1) On vérifie que $s^\nu \in \Gamma^*(\rho, y^\nu) \cap \Gamma(Q_{y^\nu}, y^\nu)$ dans le cas (a) comme dans le cas (b).

Propriétés :

Si on pose $v^\nu = y^\nu + s^\nu$:

$$v^\nu = \begin{cases} x^\nu & \text{si on est dans le cas (a)} \\ x^\nu + \rho^\nu s^{\nu-1} & \text{si on est dans le cas (b)} \end{cases}$$

alors v^ν vérifie les propriétés suivantes :

$$(5.3') \quad ((y^{\nu+1} | v^\nu - y^{\nu+1})) \geq 0$$

$$(5.4') \quad ((v^\nu - z | y^\nu)) \leq 0 \quad \forall z \in P \cap K^\nu$$

La propriété (5.2) est encore vérifiée par y^ν et $y^{\nu+1}$.

Remarque :

$v^\nu \in H(y^\nu) \cap P$ mais on peut avoir $v^\nu \notin K^\nu$.

D'où l'importance de l'inégalité $\rho < \tau$ pour que la suite $\{v^\nu - y^\nu\}$ soit bornée.

II-CONVERGENCE-

Proposition 5.1

"L'algorithme n°4 converge vers la solution du problème (P1)".

Démonstration (cf Haugazeau [14]).

Soit $y^0 \in P$. Nous allons montrer que la suite $\{y^\nu\}$, obtenue au moyen de l'algorithme n°4, converge vers la solution y_σ .

a) étude de $F(y^\nu - y_\sigma)$

$$F(y^\nu - y_\sigma) = ((y^\nu - y_\sigma | y^\nu - y_\sigma)) \leq ((y^\nu | y^\nu - y_\sigma))$$

d'après la proposition 2.2 qui permet de caractériser y_σ .

Si $y_\sigma \in K^v$, alors la propriété (5-4) entraîne :

$$((y^v | y^v - y_\sigma)) \leq ((y^v | y^v - x^v)).$$

si $y_\sigma \notin K^v$, alors $y^v + \frac{r}{\|y_\sigma - y^v\|} (y_\sigma - y^v) \in P \cap K^v$ et la propriété

(5-4) entraîne encore :

$$((x^v | y^v)) \leq ((y^v + \frac{r}{\|y_\sigma - y^v\|} (y_\sigma - y^v) | y^v))$$

soit :

$$\frac{r}{\|y_\sigma - y^v\|} ((y^v - y_\sigma | y^v)) \leq ((y^v | y^v - x^v)).$$

La suite $\{F(y^v)\}$ étant décroissante, la suite $\{y^v\}$ est bornée (cf démonstration de la proposition 2.4). Par suite il existe un entier M ($M > 1$) tel que :

$$((y^v - y_\sigma | y^v)) \leq M ((y^v | y^v - x^v)).$$

et, pour tout v , on a finalement :

$$(5-5) \quad F(y^v - y_\sigma) \leq M ((y^v | y^v - x^v)).$$

b) étude de la suite $\{((y^v | y^v - x^v))\}$.

$$\begin{aligned} ((y^v | y^v - x^v)) &= ((y^v - y^{v+1} | y^v - x^v)) + ((y^{v+1} | y^v - y^{v+1})) + ((y^{v+1} | y^{v+1} - x^v)) \\ &\leq ((y^v - y^{v+1} | y^v - x^v)) + ((y^{v+1} | y^v - y^{v+1})) \end{aligned}$$

d'après la propriété (5-3)

$$(5-6) \quad \leq \sqrt{F(y^v - y^{v+1}) F(y^v - x^v)} + ((y^{v+1} | y^v - y^{v+1}))$$

d'après l'inégalité de Schwarz.

Or l'inégalité (5-3) et la relation :

$$F(y^v) - F(y^{v+1}) = F(y^{v+1} - y^v) + 2((y^{v+1} | y^v - y^{v+1}))$$

entraînent que les suites $\{F(y^{v+1}-y^v)\}$ et $\{((y^{v+1}|y^v-y^{v+1}))\}$ convergent vers 0 puisque $\{F(y^v)-F(y^{v+1})\}$ converge vers 0.

Comme $\|y^v-x^v\| \leq R$, la suite $\{F(y^v-x^v)\}$ est bornée. D'après l'inégalité (5-6) on en déduit que la suite $\{((y^v|y^v-x^v))\}$ converge vers 0.

c) Conclusion

On a alors, d'après l'inégalité (5-5) : $\lim_{v \rightarrow \infty} F(y^v - y_\sigma) = 0$.

La suite $\{y^v\}$ étant bornée. On peut en extraire une sous-suite convergente qui converge vers $\bar{y} \in P$.

$$F(y_\sigma - \bar{y}) = 0 \implies y_\sigma - \bar{y} \in \mathcal{N}(B) \implies F(y_\sigma) = F(\bar{y}).$$

Par conséquent :

$$\lim_{v \rightarrow \infty} F(y^v) = F(\bar{y}) = d$$

et la suite $\{y^v\}$ est une suite minimisante.

On démontrerait de la même manière que la suite $\{y^v\}$ obtenue par l'algorithme n° 4' est une suite minimisante. En effet, la suite $\{y^v\}$ construite par cet algorithme vérifie toutes les propriétés de la suite $\{x^v\}$, construite par l'algorithme n° 4, dont on a besoin dans la démonstration ci-dessus.

On peut donc énoncer la proposition suivante :

Corollaire 5.1.

"L'algorithme n° 4' converge vers la solution du problème (P1)".

IV MISE EN OEUVRE DES ALGORITHMES DANS LE CAS DU PROBLEME (P1') ET PROCEDURES ALGOL.

4.1. A chaque itération v , pour les algorithmes n° 4 et 4', nous avons à trouver $x^v \in H(y^v) \cap K^v \cap P$, soit encore à minimiser la forme linéaire $x \rightarrow (x | \Omega y^v)$ sur le convexe $K^v \cap P$. Si K^v est un polyèdre convexe, nous avons à résoudre un problème de programmation linéaire.

Cas particulier du problème (P1')

Supposons que K^v soit un pavé borné de \mathbb{R}^n . Alors $P \cap K^v$ est un pavé borné de \mathbb{R}^n . Posons :

$$P \cap K^v = \{x \in \mathbb{R}^n \mid \alpha_i^v \leq x_i \leq \beta_i^v, i = 1, \dots, n\}.$$

On peut alors montrer le résultat suivant (cf Laurent [16]) :

Lemme 5.2 "Une condition nécessaire et suffisante pour que $\bar{x} \in P \cap K^v$ minimise la fonctionnelle $x \rightarrow (x | \Omega y^v)$ sur $P \cap K^v$ est que l'on ait ;

$$\Gamma^*(P \cap K^v, \bar{x}) \cap \Gamma(Q, \bar{x}) = \emptyset$$

avec $Q = \{z \in \mathbb{R}^n \mid (z | \Omega y^v) < (\bar{x} | \Omega y^v)\}$ "

Utilisons les notations introduites au chapitre III :

$$I_+(\bar{x}) = \{i \in \{1, \dots, n\} \mid \bar{x}_i = \beta_i^v\}, \quad I_-(\bar{x}) = \{i \in \{1, \dots, n\} \mid \bar{x}_i = \alpha_i^v\}$$

On a alors :

$$\Gamma^*(P \cap K^v, \bar{x}) = \{s \in \mathbb{R}^n \mid s_i \geq 0 \text{ si } i \in I_-(\bar{x}), s_i \leq 0 \text{ si } i \in I_+(\bar{x})\}$$

$$\Gamma(Q, \bar{x}) = \{s \in \mathbb{R}^n \quad (s \quad \Omega y^\nu) < 0\},$$

et on déduit aisément du lemme ci-dessus que :

" Une condition nécessaire et suffisante pour que $\bar{x} \in H(y^\nu) \cap P \cap K^\nu$ est que l'on ait :

$$\underline{\bar{x}_i = \alpha_i^\nu \quad \text{si} \quad (\Omega y^\nu)_i > 0 \quad \text{et} \quad \bar{x}_i = \beta_i^\nu \quad \text{si} \quad (\Omega y^\nu)_i < 0 "}$$

4.2. Procédure FRANCK WOLFE.

Cette procédure résoud le problème (Pl') dans le cas où le convexe P est borné. Elle utilise l'algorithme de Franck et Wolfe ; à chaque itération on a $P \cap K^\nu = P$.

Etant donné y^ν , on définira x^ν par :

$$x_i^\nu = a_i \quad \text{si} \quad (\Omega y^\nu)_i > 0$$

$$x_i^\nu = b_i \quad \text{si} \quad (\Omega y^\nu)_i < 0$$

$$x_i^\nu = \frac{a_i + b_i}{2} \quad \text{si} \quad (\Omega y^\nu)_i = 0$$

Remarque :

Si on a $(\Omega y^\nu)_i = 0$ pour un indice i, on a une infinité de choix possibles pour x_i^ν .

Notice d'emploi :

FRANCK WOLFE(N, A, B, Y, PETI, EPS, GRAD).

Etant donné l'entier N, le pavé P de \mathbb{R}^N est défini par les tables A[1 : N] et B[1 : N]. La procédure GRAD(Y, N, MU) calcule au point $y = Y[1 : N]$ le vecteur $\Omega y = MU[1 : N]$.

Il y a deux tests d'arrêt :

$$\text{si } \frac{|F(y^\nu) - F(y^{\nu+1})|}{F(y^\nu)} < \text{PETI} \text{ ou si } |(x^\nu - y^\nu | \Omega y^\nu)| < \text{EPS}$$

on arrête les calculs

4.3. Procédure FRANCÓNJ.

Cette procédure résoud le problème (P') au moyen des algorithmes n°4 ou n°4'.

Notice d'emploi.

FRANCÓNJ(N, M, A, C, Y, R, RO, EPS, EPS1, MINLIN, GRAD, L1).

Etant donnés N et M, on définit le polyèdre P par :

$$P = \{X \in \mathbb{R}^N \mid \sum_{I=1}^N A[J, I] \times X[I] \leq C[J], \quad J = 1, \dots, M\} .$$

A partir de y^0 contenu dans $Y[1 : N]$ à l'appel de la procédure on construit une suite y^ν , contenue à chaque itération dans Y, au moyen des algorithmes n°4 ou 4' : 4 si le booléen L1 contient la valeur "faux", 4' sinon. A chaque itération ν le voisinage K^ν est défini par :

$$K^\nu = \{X \in \mathbb{R}^N \mid \text{ABS}(X[I] - Y[I]) \leq R\} .$$

Dans l'algorithme 4, on choisira le point v^ν de sorte que $\|v^\nu - x^\nu\| < RO..$

La procédure GRAD Y, N, MU) calcule au point $y = Y$ le vecteur $\Omega y = MU[1 : N]$.

La procédure MINLIN(N, M, A, C, L, R, Y, MU, Z) calcule le point $x = Z[1 : N]$ qui minimise la fonction $(x \quad z)$ ($z = MU[1 : N]$) sur le polyèdre $P \cap K$ où P est défini par M, A et C et K par Y et R.

La procédure comporte deux tests d'arrêts :

$$\text{si } \frac{F(y^\nu) - F(y^{\nu+1})}{F(y^\nu)} < \text{EPS} \text{ ou si } (x^\nu - y^\nu | \Omega y^\nu) < \text{EPS1}$$

on arrête les calculs.

Nous donnons aussi la procédure MINLIN dans le cas particulier du problème (P1').

```

PROCEDURE FRANCONJ(N,M,A,C,Y,R,RC,EPS,EPS1,MINLIN,GRAD,L1);
INTEGER N,M;
REAL R,RO,EPS,EPS1;
BOOLEAN L1;
ARRAY A,C,Y;
PROCEDURE MINLIN,GRAD;

```

```

BEGIN
  INTEGER I,J;
  BOOLEAN L;
  REAL F,F1,S1,S2,D,DP,E,CY,CS;
  ARRAY Z,U,S,MU,MUPR(1:N/1);
  L:=FALSE;
  F:=0;
  GOID TEST;
  DIRECTION:=MINLIN(N,M,A,C,R,Y,MU,Z);
  E:=0;
  FOR I:=1 STEP 1 UNTIL N DO
    BEGIN
      S(I/1):=Z(I/1)-Y(I/1);
      F:=F+S(I/1)*MU(I/1);
    END;
  IF F<EPS1 THEN GOID TERM;
  IF L THEN GOID PAS;
  GRADIN,U,MUPR;
  S1:=S2:=0;
  FOR I:=1 STEP 1 UNTIL N DO
    BEGIN
      S2:=S2+MUPR(I/1)*S(I/1);
      S1:=S1+MUPR(I/1)*U(I/1);
    END;
  D:=-S2/S1;
  IF ABS(D)>RO THEN D:=RC*SIGN(D);
  FOR I:=1 STEP 1 UNTIL M DO
    BEGIN
      CY:=CS:=0;
      FOR J:=1 STEP 1 UNTIL N DO
        BEGIN
          CY:=CY+A(I,J/1)*Z(I/1);
          CS:=CS+A(I,J/1)*U(I/1);
        END;
      IF CS>=F THEN DP:=(C(I/1)-CY)/CS;
      IF ABS(DP)<ABS(D) THEN D:=DP;
    END;
  FOR I:=1 STEP 1 UNTIL N DO S(I/1):=S(I/1)+U(I/1)*D;
  PAS:F:=0;

```



```

FOR I:=1 STEP 1 UNTIL N DO E:=E+S(/I/)*S(/I/);
E:=SQRT(E);
FOR I:=1 STEP 1 UNTIL N DO S(/I/):=S(/I/)/E;
DP:=0;
FOR I:=1 STEP 1 UNTIL M DO
  BEGIN
    CY:=CS:=0;
    FOR J:=1 STEP 1 UNTIL N DO
      BEGIN
        CY:=CY+A(/I,J/)*Y(/J/);
        CS:=CS+A(/I,J/)*S(/J/);
      END ;
    IF (CS>'-8) & (C(/I/)-CY>'-6) THEN
      BEGIN
        D:=(C(/I/)-CY)/ABS(CS);
        IF (DP=0) | (D<DP AND DP=-0) THEN DP:=0
      END ;
    END ;
    CHOIXDUPAS: GRAD(N,S,MUPR);
    S1:=S2:=0;
    FOR I:=1 STEP 1 UNTIL N DO
      BEGIN
        S1:=S1+MUPR(/I/)*S(/I/);
        S2:=S2+MU(/I/)*S(/I/);
      END ;
    D:=-S2/S1;
    IF (DP=-0) & (D<DP) THEN
      BEGIN
        L:= TRUE ;
        FOR I:=1 STEP 1 UNTIL N DO
          BEGIN
            Y(/I/):=Y(/I/)+S(/I/)*D;
            U(/I/):=S(/I/);
          END ;
        GO TO TEST
      END ;
    L:= FALSE ;
    FOR I:=1 STEP 1 UNTIL N DO Y(/I/):=Y(/I/)+S(/I/)*DP;
    TEST: GRAD(N,Y,MU);
    F1:=0;
    FOR I:=1 STEP 1 UNTIL N DO F1:=F1+Y(/I/)*MU(/I/);
    IF ABS((F1-F)/F1)<EPS THEN GO TO TERM;
    F:=F1;
    L:=L&L1;
    GO TO DIRECTION;
  TERM:
  END FRANCONJ;
END ;

```



```
PROCEDURE MINLIN(N,M,A,C,R,Y,MU,Z);
```

```
INTEGER N,M;
```

```
ARRAY A,C,Y,MU,Z;
```

```
REAL R;
```

```
BEGIN
```

```
INTEGER I,J,K,KO;
```

```
FOR I:=1 STEP 1 UNTIL N DO Z(I/):= IE MU(I/)>'-8
```

```
THEN Y(I/)-R ELSE IE MU(I/)<'-8 THEN Y(I/)+R
```

```
ELSE Y(I/);
```

```
FOR I:=1 STEP 1 UNTIL N DO
```

```
  BEGIN
```

```
    FOR J:=1 STEP 1 UNTIL N DO IE ABS(A(I,J/))>'-8
```

```
    THEN
```

```
      BEGIN
```

```
        K:=J;
```

```
        KO:=SIGN(A(I,J/));
```

```
        GOTO JO
```

```
      END ;
```

```
    JO: IE Z(I/K/)*KO>=C(I/) THEN Z(I/):=C(I/)*KO;
```

```
  END ;
```

```
END MINLIN;
```



```

PROCEDURE FRANCKWOLFE(N,A,B,Y,PETI,EPS,GRAD);
INTEGER N;
ARRAY A,B,Y;
REAL PETI, EPS;
PROCEDURE GRAD;

```

```

BEGIN

```

```

    INTEGER I;

```

```

    REAL S,T,D,F,F1;

```

```

    REAL ARRAY Z,MU,MUPR(/1:N/);

```

```

    GRAD(N,Y,MU);

```

```

    F:=0;

```

```

    FOR I:=1 STEP 1 UNTIL N DO F:=F+MU(/I/)*Y(/I/);

```

```

    ITER: FOR I:=1 STEP 1 UNTIL N DO IF MU(/I/)>=-8

```

```

    THEN Z(/I/):=A(/I/) ELSE IF MU(/I/)<=-8 THEN Z(/I/):=B(/I/)

```

```

    ELSE Z(/I/):=(A(/I/)+B(/I/))/2;

```

```

    S:=0;

```

```

    FOR I:=1 STEP 1 UNTIL N DO S:=S+(Z(/I/)-Y(/I/))*MU(/I/);

```

```

    IF ABS(S)<EPS THEN GOTO TERM;

```

```

    GRAD(N,Z,MUPR);

```

```

    T:=0;

```

```

    FOR I:=1 STEP 1 UNTIL N DO T:=T+(Z(/I/)-Y(/I/))*MUPR(/I/);

```

```

    IF T<0 THEN

```

```

        BEGIN

```

```

            FOR I:=1 STEP 1 UNTIL N DO Y(/I/):=Z(/I/);

```

```

            GOTO APPUI

```

```

        END ;

```

```

    D:=T/(T-S);

```

```

    FOR I:=1 STEP 1 UNTIL N DO Y(/I/):=D*Y(/I/)+(1-D)*Z(/I/);

```

```

    APPUI: GRAD(N,Y,MU);

```

```

    F1:=0;

```

```

    FOR I:=1 STEP 1 UNTIL N DO F1:=F1+Y(/I/)*MU(/I/);

```

```

    IF ABS((F-F1)/F)<PETI THEN GOTO TERM ELSE

```

```

        BEGIN

```

```

            F:=F1;

```

```

            GOTO ITER

```

```

        END ;

```

```

    TERM:

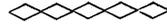
```

```

END FRANCKWOLFE;

```


C H A P I T R E VI



METHODE DES DIRECTIONS CONJUGUEES

Nous exposons dans ce chapitre une méthode due à Zoutendijk [24]. Cette méthode résoud le problème (P1). Elle présente l'originalité de converger avec un nombre fini d'itérations.

Dans un premier paragraphe nous définissons la notion "directions conjuguées" et nous donnons les propriétés de ces directions conjuguées. L'algorithme décrit dans le deuxième paragraphe est fondé sur ces propriétés. Au troisième paragraphe nous étudions certains aspects pratiques de l'algorithme et nous donnons une procédure Algol.

I-DIRECTIONS CONJUGUEES-

Définition.

Soit $\{ s^1, \dots, s^k \}$ une suite finie d'éléments de \mathbb{R}^n qui vérifient la propriété :

$$(6.0) \quad \begin{aligned} (s^i \mid \Omega s^j) &= 0 && \text{si } i \neq j \\ (s^i \mid \Omega s^i) &\neq 0 && \text{si } i \leq \min(k, n). \end{aligned}$$

Ces k vecteurs sont appelés "directions conjuguées".

Notations.

On pose $((x \mid y)) = (x \mid \Omega y) = x^T \Omega y \quad \forall x, y \in \mathbb{R}^n$.

$$S(s^1, \dots, s^k) = \{ x \in \mathbb{R}^n \mid ((x \mid s^i)) = 0 \quad i = 1, \dots, k \}$$

Propriétés.

Pour tout $l \leq n$, les l premiers vecteurs, s^1, \dots, s^l , sont linéairement indépendants : en effet si $\sum_{i=1}^l \lambda_i s^i = 0$, la propriété (6.0) entraîne immédiatement $\lambda_i = 0$ pour tout $i \in \{1, \dots, l\}$.

Si $k > n$, on a alors : $s^{n+1} = \dots = s^k = 0$, soit encore : $S(s^1, \dots, s^l) = 0$ pour tout $l \geq n$.

Soit $\{x^k, k=1, \dots, \nu\}$ une suite d'éléments de P définis par les relations

$$(6.1) \quad x^{k+1} = x^k + \lambda''(x^k, s^k) s^k$$

$$(6.2) \quad \lambda''(x^k, s^k) < \lambda'(x^k, s^k)$$

$$(6.3) \quad s^k \in S(s^1, \dots, s^{k-1}).$$

On a la propriété suivante :

Proposition 6.1

"Si $\Gamma^*(P, x^\nu) \cap \Gamma(Q_{x^\nu}, x^\nu) \cap S(s^1, \dots, s^{\nu-1}) = \emptyset$ alors x^ν est la solution du problème (P1)".

Démonstration

Supposons que x^ν ne soit pas solution. Alors $\Gamma^*(P, x^\nu) \cap \Gamma(Q_{x^\nu}, x^\nu) \neq \emptyset$. Soit $s \in \Gamma^*(P, x^\nu) \cap \Gamma(Q_{x^\nu}, x^\nu)$.

On pose $t = s - \sum_{k=1}^{\nu-1} \mu_k s^k$ avec $\mu_k = \frac{((s | s^k))}{((s^k | s^k))} \quad \forall k \in \{1, \dots, \nu-1\}$

$$a \quad t \in S(s^1, \dots, s^{\nu-1}).$$

En effet, $((t \mid s^k)) = ((s \mid s^k)) - \mu_k((s^k \mid s^k))$ pour tout $k \in \{1, \dots, \nu-1\}$ et ceci par définition de s^k (propriété (6.3)). Donc $((t \mid s^k)) = 0$ pour tout $k \in \{1, \dots, \nu-1\}$.

b Pour tout $k \in \{1, \dots, \nu-1\}$, on a :

$$x^\nu = x^k + \sum_{l=k}^{\nu-1} \lambda''(x^l, s^l) s^l.$$

La définition de $\lambda''(x^l, s^l)$ entraîne alors :

$$((s^l \mid x^k)) = ((s^l \mid s^l)) + \lambda''(x^l, s^l) ((s^l \mid s^l)) = 0 \text{ pour tout } l \in \{k, \dots, \nu-1\}.$$

Par conséquent $((s \mid s^k)) = ((t \mid x^k))$ et $t \in \Gamma(\underline{\mathbb{Q}}_k, \underline{x^k})$.

(c) Soit $I = I(x^\nu)$. La propriété (6.2) entraîne :

$$I(x^k) \supset I \quad \forall k \in \{1, \dots, \nu-1\}.$$

$$\begin{aligned} \text{Donc } \sum_{j=1}^n a_i^j s_j &= \sum_{j=1}^r a_i^j t_j + \sum_{k=1}^{\nu-1} \mu_k \left(\sum_{j=1}^n a_i^j s_j^k \right) \\ &= \sum_{j=1}^n a_i^j t_j \quad \forall i \in \{1, \dots, m\} \end{aligned}$$

Comme $s \in \Gamma^*(P, x^k)$, $\sum_{j=1}^n a_i^j s_j \leq 0$ pour tout $i \in I$ et l'égalité ci-dessus nous donne : $t \in \Gamma^*(P, x^\nu)$ d'où la contradiction.

II-ALGORITHME N°5-

Notations

Soit $J \subset \{1, \dots, m\}$

$$P_J = \{x \in \mathbb{R}^n \mid x \in P, \sum_{i=1}^n a_j^i x_i = c_j \quad \forall j \in J\}$$

P_J est une face du polyèdre convexe P .

2.1 Description de l'algorithme N°5.

$$x^\nu = x^{\nu-1} + \lambda^{\nu-1} s^{\nu-1}$$

$$\text{avec } \lambda^{\nu-1} = \min(\lambda''(x^{\nu-1}, s^{\nu-1}), \lambda'(x^{\nu-1}, s^{\nu-1}))$$

Soit $J \subset \{1, \dots, m\}$. J peut être vide. Soit r un indice, $r \leq \nu - 1$.

Nous allons déterminer la direction de descente s^ν .

$$\textcircled{a} \quad \underline{\lambda^{\nu-1} = \lambda''(x^{\nu-1}, s^{\nu-1}) < \lambda'(x^{\nu-1}, s^{\nu-1})}$$

s^ν est la projection de $-\Omega x^\nu$ sur le cône :

$$C^\nu = \Gamma^*(P_J, x^\nu) \cap S(s^\nu, \dots, s^{\nu-1}).$$

$$\textcircled{b} \quad \underline{\lambda^{\nu-1} = \lambda'(x^{\nu-1}, s^{\nu-1})}$$

Supposons que $i \in I^\nu \cap I^k$ avec $k < \nu$, c'est-à-dire que x^ν soit le deuxième élément de la suite construite par l'algorithme qui appartienne à la face $P_{\{i\}}$

On ajoute i à l'ensemble J et on pose $r = \nu$.

s^ν est la projection de $-\Omega x^\nu$ sur le cône $\Gamma^*(P_J, x^\nu)$.

$$\textcircled{c} \quad \text{Si } s^\nu = 0, \text{ on a alors :}$$

Soit $(-\Omega x^\nu | y) \leq 0 \quad \forall y \in \Gamma^*(P_J, x^\nu) \cap S(s^\nu, \dots, s^{\nu-1})$ et d'après la proposition

6.1 d'une part, le lemme 2.1 d'autre part, x^ν est un point de P_J qui minimise F ;

Soit $(-\Omega x^\nu | y) \leq 0 \quad \forall y \in \Gamma^*(P_J, x^\nu)$ et x^ν est encore un point de P_J qui

minimise F .

Si $J = \emptyset$, x^ν est la solution

Si $J \neq \emptyset$, on pose $J = \emptyset$ et on prend pour direction de descente

$s^\nu = \tilde{s}^\nu$ où \tilde{s}^ν désigne la projection de $-\Omega x^\nu$ sur le cône $\Gamma^*(P, x^\nu)$ si $x^{\nu-1}$ vérifie le cas (b), sur le cône $\Gamma^*(P, x^\nu) \cap S(s^r, \dots, s^{\nu-1})$ si $x^{\nu-1}$ vérifie (a).

(d) Si x^ν n'est pas solution de (P1), on pose :

$$x^{\nu+1} = x^\nu + \lambda^\nu s^\nu$$

2.2. Propriétés.

1) Comme s^ν est la projection de $-\Omega x^\nu$ sur un cône, s^ν et x^ν vérifient :

$$0 = (s^\nu + \Omega x^\nu | s^\nu) = ((s^\nu | x^\nu)) + \|s^\nu\|^2.$$

Donc $((s^\nu | x^\nu)) = -\|s^\nu\|^2 < 0$ et $s^\nu \in \Gamma^*(Q_\nu, x^\nu)$.

Par définition de λ^ν , on a alors $\underline{F(x^{\nu+1})} < \underline{F(x^\nu)}$.

2) Comme la suite $\{F(x^\nu)\}$ est bornée, la suite $\{\lambda^\nu\}$ est bornée. On en déduit que si $\lambda^\nu = \lambda^\nu(x^\nu, s^\nu)$, on a forcément $((s^\nu | s^\nu)) \neq 0$. Les directions s^r, \dots, s^ν sont bien alors des directions conjuguées.

2.3. Convergence.

Proposition 6.2

" L'algorithme n°5 converge en un nombre fini d'itérations ".

Démonstration :

Le polyèdre convexe P a un nombre fini de faces. Soit $J \in \{1, \dots, m\}$. Si $x^\nu \in P_J$ et si J contient m_J éléments, après $2m_J + (n-1)m_J$ c'est-à-dire $(n+1) \times m_J$ itération au plus, on obtient le minimum de F sur la face P_J . D'où le résultat.

III-ASPECTS PRATIQUES DE L'ALGORITHME N°5 ET PROCEDURES ALGOL-

3.1 Détermination de la direction de descente.

A chaque itération ν , la direction de descente s^ν est la projection de $-\Omega y^\nu$ sur un cône Γ^ν .

$$\Gamma^\nu = \Gamma^*(P_J, x^\nu) \quad \text{si } r = 0$$

$$= \Gamma^*(P_J, x^\nu) \cap S(s^1, \dots, s^r) \quad \text{si } r \geq 1.$$

Posons $I^\nu = \{j \in \{1, \dots, m\} \mid \sum_{i=1}^n a_j^i x_i^\nu = c_j\}$. Par définition, $J \subset I^\nu$.

Comme $P_J = \{x \in P \mid \sum_{i=1}^n a_j^i x_i = c_j, \quad j \in J\}$ on vérifie aisément que :

$$\Gamma^*(P_J, x^\nu) = \{s \in \mathbb{R}^n \mid \sum_{i=1}^n a_j^i s_i = 0 \text{ si } j \in J, \sum_{i=1}^n a_j^i s_i \leq 0 \text{ si } j \in I^\nu\}$$

Le cône Γ^ν est donc défini par une intersection finie de demi espaces fermés. Nous déterminerons alors s^ν au moyen de l'algorithme d'éclatement des contraintes (cf Lions et Tenau [20], Haugazeau [13,14], Laurent et Martinet [18])

Principe de l'algorithme d'éclatement des contraintes.

Soit Δ un convexe fermé non vide d'un espace de Hilbert H . On suppose que Δ est donné sous la forme d'une intersection finie de demi-espaces fermés :

$$\Delta = \bigcap_{i \in I} \Delta_i$$

L'algorithme permet d'obtenir la projection \bar{u} d'un point $u \notin \Delta$ sur Δ .

La méthode est fondée sur la formule de dualité suivante :

$$d = \min (\|v - u\| \mid v \in \Delta)$$

$= \max[\min(\|v - u\| \mid v \in D) \mid D \supset \Delta]$ où D désigne un demi-espace fermé de H .

On construit une suite de demi espaces fermés $D^l \supset \Delta$ tels que

$$\lim_{l \rightarrow \infty} \min (\|v - u\|, v \in D^l) = d$$

$$\lim_{l \rightarrow \infty} u^l = \bar{u} \quad \text{avec } u^l \text{ projection de } u \text{ sur } D^l$$

Algorithme : Soit u^l tel que $D^l = \{x \in H \mid (u^l - u \mid x - u^l) \geq \Delta\}$ contienne Δ .

Si $u^l \in \Delta_j$ pour tout $i \in I$, alors $u^l = \bar{u}$. Supposons que $u^l \notin \Delta_i^l$ et appelons u^{l+r} la projection de u sur $D^l \cap \Delta_i^l$. Considérons :

$$D^{l+1} = \{x \in H \mid (u^{l+1} - u \mid x - u^{l+1}) \geq 0\}.$$

Alors D^{l+1} contient Δ .

On pose $m_j^l = \min(\|v - u^l\| \mid v \in \Delta_j^l)$. On montre que si on choisit i^l tel que :

$$m_{i^l}^l = \max(m_i^l \mid i \in I)$$

alors la suite $\{u^l\}$ converge vers \bar{u} dans H .

Remarque :

Nous n'avons donné ici que l'une des versions les plus simples de l'algorithme.

Calcul de u^{l+1} : soit \tilde{u}^{l+1} la projection de u sur Δ_i^l .

Deux cas sont possibles :

$$\textcircled{1} \quad \tilde{u}^{l+1} \in D^l ; \text{ alors } \tilde{u}^{l+1} = u^{l+1}$$

$$\textcircled{2} \quad \tilde{u}^{l+1} \notin D^l. \text{ Alors } u^{l+1} \in \mathcal{F}_r(D^l) \cap \mathcal{F}_r(\Delta_i^l) \text{ où } \mathcal{F}_r(D^l)$$

(resp. $\mathcal{F}_r(\Delta_i^l)$) désigne la frontière du demi espace fermé D^l (resp. Δ_i^l).

3.2. Notice d'emploi de la procédure GRADCØNJ.

Cette procédure résoud le problème (P') au moyen de l'algorithme n°5.

GRADCØND(N, M, Y, A, C, RO, TO, EP, EPS, GRAD, ECLATCONT)

Etant donnés les entiers N et M, le polyèdre P est défini par les tableaux A[1 : M, 1 : N] et C[1 : M].

$$P = \{X \in \mathbb{R}^N \mid \sum_{I=1}^N A[J,I] \times X[I] \leq C[J], \quad J = 1, \dots, M\}$$

Le réel RO est à déterminer l'ensemble I^ν : étant donné le point $y^\nu = Y[1 : N]$ si $|C[J] - \sum_{I=1}^N A[J,I] \times Y[I]| < RO$ alors $J \in I^\nu$.

Etant donné le point $y^\nu = Y$, la procédure GRAD(Y,N,MU) calcule le vecteur $\Omega y^\nu = MU[1 : N]$.

Etant donné le point $\Omega y^\nu = MU$, la procédure ECLATCONT calcule la projection s^ν de $-\Omega y^\nu$ sur le cône Γ^ν . Le réel TO fixe la fonction de cette procédure..

Si $\|s^\nu\| < EP$ et si $J = \emptyset$ alors les calculs s'arrêtent; si $J \neq \emptyset$, on recommence la détermination de s^ν (cf partie (c) de l'algorithme).

La procédure comporte en outre un second test d'arrêt :

$$\text{si } \left| \frac{F(y^\nu) - F(y^{\nu+1})}{F(y^\nu)} \right| < EPS, \text{ on arrête les calculs.}$$

Procédure GRADCØNJSIMPLE.

Cette procédure résoud le problème (P1') : elle a la même structure que la procédure GRADCONJ.

GRADCØNJSIMPL (N, Y, A, B, RØ, TØ, EPS, GRAD, ECLATCONT)

Le polyèdre P est défini dans ce cas par l'entier N et les tableaux A[1 : N] et B[1 : N].

$$P = \{X \in \mathbb{R}^N \mid A[I] \leq X[I] \leq B[I], \quad I = 1, \dots, N\}$$

3.3. Notice d'emploi de la procédure ECLATCØNJ.

ECLATCØNT(N, M, Y, A, S, FR, VAR, CØNJ, R, E).

Etant donné l'entier N, les tableaux entiers VAR[1 : N] et FR[1 : N] qui contiennent des 0 ou des 1 définissent le sous-espace vectoriel Γ_1 et le cône Γ_2 :

$$\Gamma_1 = \{X \in \mathbb{R}^N \mid \sum_{I=1}^N A[J,I] \times X[I] = 0 \text{ si } VAR[J] = 1\}$$

$$\Gamma_2 = \{X \in \mathbb{R}^N \mid \sum_{I=1}^N A[J,I] \times X[I] \leq 0 \text{ si } FR[J] = 1\}$$

Le tableau CØNJ[1 : R, 1 : N] définit un sous espace vectoriel Γ_3 :

$$\Gamma_3 = \{x \in \mathbb{R}^n \mid \sum_{I=1}^N CØNJ[K,I] \times X[I] = 0 \quad K = 1, \dots, R\}.$$

La procédure que nous donnons calcule la projection $s = S[1 : N]$ du vecteur $-y$, $y = Y[1 : N]$ sur le cône $\Gamma = \Gamma_1 \cap \Gamma_2$ si $R = 0$, sur le cône $\Gamma = \Gamma_1 \cap \Gamma_2 \cap \Gamma_3$ si $R > 0$, au moyen de l'algorithme d'éclatement des contraintes. Elle résoud le cas particulier du problème (P1') (si on veut l'utiliser avec GRADCØNJSIMPL, il suffit de faire : "M := N"). Elle s'utilise avec GRADCØNJ.

On arrête les calculs dès que s est à une distance inférieure ou égale à E de chacun des demi espaces dont l'intersection définit le cône Γ .


```

PROCEDURE GRADCCNJSIMPL(N,Y,A,B,RO,TO,FP,FPS,GRAD,ECLATCONT);
INIEGER N;
ARRAY Y,A,B;
REAL RO,TO,FP,FPS;
PROCEDURE GRAD,ECLATCONT;

```

```

BEGIN

```

```

INIEGER I,J,V,R;
REAL FCTO,FCT1,D,DP,F,E;
REAL ARRAY MU,S,Y1,MU1(/1:N/),CONJ,H(/1:N,1:N/);
INIEGER ARRAY FR,FR1,FR2,VAR(/1:N/);
FOR I:=1 STEP 1 UNIL N DO FOR J:=1 STEP 1 UNIL N
DO H(/I,J/):= IE I=J IHEN 1 ELSE 0;
FCTO:=FCT1:=0;
FOR I:=1 STEP 1 UNIL N DO
BEGIN
FR2(/I/):=FR1(/I/):=VAR(/I/):=0;
FOR J:=1 STEP 1 UNIL N DO CONJ(/I,J/):=0;
FR(/I/):= IE Y(/I/)-A(/I/)<RO IHEN -1 ELSE IE B(/I/)-Y(/I/)<
RO IHEN 1 ELSE 0;
END ;
GRAD(N,Y,MU);
LAMB1:R:=0;
ECLATCONT(N,A,MU,F,S,FR,VAR,CONJ,R,TO);
LAMB2:FCTO:=0;
FOR I:=1 STEP 1 UNIL N DO FCTO:=FCTO+MU(/I/)*Y(/I/);
IE ABS(FCTO-FCT1)/FCTO<EPS IHEN
BEGIN
IE R=0 IHEN GOIQ ARRET ELSE
BEGIN
R:=0;
ECLATCONT(N,N,MU,H,S,FR,VAR,CONJ,R,TO);
END ;
END ;
F:=0;
V:=0;
FOR I:=1 STEP 1 UNIL N DO
BEGIN
F:=F+S(/I/)*S(/I/);
V:=V+VAR(/I/)*VAR(/I/);
END ;
IE F<EPS IHEN
BEGIN
IE V=0 IHEN GOIQ ARRET;
FOR I:=1 STEP 1 UNIL N DO VAR(/I/):=0;
END ;

```



```

CHOIXDLPAS;DP:=0;
F:=0;
FOR I:=1 STEP 1 UNTIL N DO
  BEGIN
    IF S(I)/I>=E THEN D:=(R(I)/I)-Y(I)/I/S(I/I) ELSE
    IF S(I)/I<=-E THEN D:=(A(I)/I)-Y(I)/I/S(I/I);
    IF (DP=0) OR (D<DP) AND (DP<=0) THEN
      BEGIN
        DP:=D;
        J:=I
      END ;
    END ;
  END ;
FOR I:=1 STEP 1 UNTIL N DO YI(I/I):=Y(I/I)+DP*S(I/I);
GRAD(N,Y,MU);
F:=0;
FOR I:=1 STEP 1 UNTIL N DO E:=F+S(I/I)*MU(I/I);
IF E<0 THEN
  BEGIN
    FOR I:=1 STEP 1 UNTIL N DO
      BEGIN
        VI(I/I):=YI(I/I);
        MUI(I/I):=MU(I/I);
        IF FR1(I/I)=0 THEN FR2(I/I):=FR1(I/I);
        FR1(I/I):=FR(I/I);
        FR(I/I):= IF Y(I/I)-A(I/I)<R0 THEN -1 ELSE
        IF B(I/I)-Y(I/I)<R0 THEN 1 ELSE 0;
        IF (FR(I/I)*FR2(I/I)=0) THEN VAR(I/I):=FR(I/I);
        IF R=0 THEN FOR J:=1 STEP 1 UNTIL R DO CCNJ(J,I/I):=0
      END ;
    GOIO LAMB1
  END ;
  IF E>0 THEN
    BEGIN
      DP:=DP*F/(F+E);
      R:=R+1;
      IF R>=N+1 THEN GOIO ARRET;
      FOR I:=1 STEP 1 UNTIL N DO
        BEGIN
          YI(I/I):=YI(I/I)+DP*S(I/I);
          CCNJ(I/R,I/I):=S(I/I)/SORT(F);
          IF R<3 THEN
            BEGIN
              IF FR1(I/I)=0 THEN FR2(I/I):=FR1(I/I);
              FR1(I/I):=FR(I/I)
            END ;
          FR(I/I):= IF Y(I/I)-A(I/I)<R0 THEN -1 ELSE
          IF B(I/I)-Y(I/I)<R0 THEN 1 ELSE 0;
        END ;
      GRAD(N,Y,MU);
      EGLATCONT(N,N,MU,H,S,FR,VAR,CCNJ,R,TC);
      GOIO LAMB2
    END ;
  ARRET;
END GRACCNJSIMPL;

```



```

PROCEDURE ECLATCONT(N,M,Y,A,S,FR,VAR,CCNJ,R,E);
VALUE Y;
INTEGER N,M,R;
REAL F;
INTEGER ARRAY FR,VAR;
ARRAY Y,A,S,CCNJ;

```

```

BEGIN
  INTEGER I,J,JX,K;
  REAL C,CX,A1,A2,B2,C1,LU,L1;
  ARRAY Y0,Z0(I:1:N/I);
  Y(I/I):=-Y(I/I);
  FOR I:=1 STEP 1 UNIL N DO
    BEGIN
      C:=0;
      FOR J:=1 STEP 1 UNIL N DO FR(J/I):=C+FR(J/I)*A(I,J/I);
    END ;
    FOR I:=1 STEP 1 UNIL N DO
      BEGIN
        Y0(I/I):= IE VAR(I/I)=0 THEN 0 ELSE IE FR(I/I)*Y(I/I)>0
          THEN 0 ELSE Y(I/I);
        Z0(I/I):=Y0(I/I)-Y(I/I)
      END ;
      IE R=0 THEN GOID TERM ELSE
        BEGIN
          CX:=F/2;
          JX:=0;
          K:=0;
          GOID CONT3
        END ;
        (IFR:CONT1): FOR I:=1 STEP 1 UNIL N DO IE ABS(VAR(I/I)*Y0(I/I))
          >CX THEN
          BEGIN
            JX:=I;
            CX:=ABS(Y0(I/I));
            K:=1
          END ;
          CONT2: FOR I:=1 STEP 1 UNIL N DO IE FR(I/I)*Y0(I/I)>CX
            THEN
            BEGIN
              JX:=I;
              CX:=ABS(Y0(I/I));
              K:=2
            END ;
            CONT3: FOR J:=1 STEP 1 UNIL R DO
              BEGIN
                C:=0;
                FOR I:=1 STEP 1 UNIL N DO C:=C+CCNJ(J,I/I)*Y0(I/I);
                IE ABS(C)>CX THEN
                  BEGIN
                    JX:=J;
                    CX:=ABS(C);
                    K:=3
                  END ;
                END ;
                IE ABS(CX)<F THEN GOID TERM;

```



```

C1:=0;
FOR I:=1 STEP 1 UNTIL N DO C1:=C1+70(I/I)*70(I/I);
A1:=C1;
IF K=2 THEN
  BEGIN
    A2:=70(I/JX/I);
    B2:=-Y(I/JX/I)
  END
ELSE
  BEGIN
    B2:=A2:=0;
    FOR I:=1 STEP 1 UNTIL N DO
      BEGIN
        A2:=A2+70(I/I)*CONJ(I/JX,I/I);
        B2:=B2-Y(I/I)*CONJ(I/JX,I/I)
      END ;
    END ;
    IF A1K=A2*B2 THEN
      BEGIN
        IF K=2 THEN
          BEGIN
            Y(I/JX/I):=0;
            Z(I/JX/I):=-Y(I/JX/I)
          END
        ELSE FOR I:=1 STEP 1 UNTIL N DO
          BEGIN
            Y(I/I):=Y(I/I)+B2*CONJ(I/JX,I/I);
            Z(I/I):=Y(I/I)-Y(I/I)
          END ;
          CX:=F/2;
          JX:=0;
          K:=0;
          GO TO ITER
        END ;
        L1:=B2*(A1-A2*A2);
        L0:=B2*(C1-A2*B2)/L1;
        L1:=-A2*C1-A1*B2/L1;
        FOR I:=1 STEP 1 UNTIL N DO
          BEGIN
            IF K=2 THEN
              BEGIN
                Y(I/I):=Y(I/I)+L0*Z(I/I);
                IF I=JX THEN Y(I/I):=Y(I/I)+L1*B2
              END
            ELSE Y(I/I):=Y(I/I)+L0*Z(I/I)+L1*B2*CONJ(I/JX,I/I);
                Z(I/I):=Y(I/I)-Y(I/I);
              END ;
              CX:=F/2;
              JX:=0;
              K:=0;
              GO TO ITER;
            TERM: FOR I:=1 STEP 1 UNTIL N DO S(I/I):=Y(I/I);
          END ECLATCONT;

```



```

PROCEDURE GRACONJ(N,M,Y,A,C,RO,TO,EP,EPS,GRAD,ECLATCONT);
INTEGER N,M;
ARRAY Y,A,C;
REAL RO,TO,EP,EPS;
PROCEDURE GRAD,ECLATCONT;

```

```

BEGIN

```

```

    INTEGER I,J,V,R;
    REAL FCTO,FCT1,D,DP,F,E,CY,CS;
    ARRAY S,Y1,MU,MU1(/1:N/),CONJ(/1:N,1:N/);
    INTEGER ARRAY FR,FR1,FR2,VAR(/1:M/);
    DEB:FCTO:=FCT1:=0;
    FOR I:=1 STEP 1 UNTIL M DO
        BEGIN
            FR2(/I/):=FR1(/I/):=VAR(/I/):=0;
            E:=0;
            FOR J:=1 STEP 1 UNTIL N DO E:=Y(/J/)*A(/I,J/)+E;
            FR(/I/):= IF C(/I/)-E<RO THEN 1 ELSE 0;
        END ;
    FOR I:=1 STEP 1 UNTIL N DO FOR J:=1 STEP 1 UNTIL N
        DO CONJ(/I,J/):=0;
    GRAD(N,Y,MU);
    LAMB1:R:=0;
    ECLATCONT(N,M,MU,A,S,FR,VAR,CONJ,R,TO);
    LAMB2:FCTO:=0;
    FOR I:=1 STEP 1 UNTIL N DO FCTO:=FCTO+MU(/I/)*Y(/I/);
    IF ABS((FCTO-FCT1)/FCTO)<EP THEN
        BEGIN
            IF R=0 THEN GOID ARRET ELSE
                BEGIN
                    R:=0;
                    ECLATCONT(N,M,MU,A,S,FR,VAR,CONJ,R,TO);
                END ;
            END ;
        FCT1:=FCTO;
        TEST1:F:=0;
        V:=0;
        FOR I:=1 STEP 1 UNTIL N DO F:=F+S(/I/)*S(/I/);
        FOR I:=1 STEP 1 UNTIL M DO V:=V+VAR(/I/)*VAR(/I/);
        IF F<EPS THEN
            BEGIN
                IF V=0 THEN FOR I:=1 STEP 1 UNTIL M DO VAR(/I/):=0;
            END ;
        CHOIXDUPAS:DP:=D:=0;
        FOR I:=1 STEP 1 UNTIL M DO
            BEGIN
                CY:=CS:=0;
                FOR J:=1 STEP 1 UNTIL N DO
                    BEGIN
                        CY:=CY+A(/I,J/)*Y(/J/);
                        CS:=CS+A(/I,J/)*S(/J/);
                    END ;
            END ;

```



```

IF ABS(CS)>*-8 THEN D:=(C(/I/)-CY)/CS;
IF (DP=0)/((D<DP)&(DP->=0)) THEN DP:=D;
END ;
FOR I:=1 STEP 1 UNTIL N DO Y1(/I/):=Y(/I/)+DP*S(/I/);
GRAD(N,Y1,MU1);
E:=0;
FOR I:=1 STEP 1 UNTIL N DO E:=E+S(/I/)*MU1(/I/);
IF E<0 THEN
BEGIN
FOR I:=1 STEP 1 UNTIL N DO
BEGIN
Y(/I/):=Y1(/I/);
MU(/I/):=MU1(/I/);
IF R->=0 THEN FOR J:=1 STEP 1 UNTIL R DO CONJ(/J,I/):=0
END ;
FOR I:=1 STEP 1 UNTIL M DO
BEGIN
IF FR1(/I/)->=0 THEN FR2(/I/):=FR1(/I/);
FR1(/I/):=FR(/I/);
E:=0;
FOR J:=1 STEP 1 UNTIL M DO E:=E+Y(/J/)*A(/I,J/);
FR(/I/):= IF C(/I/)-E<R0 THEN 1 ELSE 0;
IF (FR(/I/)*FR2(/I/)->=0) THEN VAR(/I/):=FR(/I/);
END ;
GOIQ LAMB1
END
ELSE IF E>0 THEN
BEGIN
DP:=DP*F/(F+E);
R:=R+1;
IF R>=N+1 THEN GOIQ ARRET;
FOR I:=1 STEP 1 UNTIL N DO
BEGIN
Y(/I/):=Y(/I/)+DP*S(/I/);
CONJ(/R,I/):=S(/I/)*SQRT(F)
END ;
IF R<3 THEN FOR I:=1 STEP 1 UNTIL M DO
BEGIN
IF FR1(/I/)->=0 THEN FR2(/I/):=FR1(/I/);
FR1(/I/):=FR(/I/);
E:=0;
FOR J:=1 STEP 1 UNTIL N DO E:=E+Y(/J/)*A(/I,J/);
FR(/I/):= IF C(/I/)-E<R0 THEN 1 ELSE 0;
END ;
GRAD(N,Y,MU);
ECLATCONT(N,M,MU,A,S,VAR,CONJ,R,TD);
GOIQ LAMB2
END ;
ARRET:
END GRADCONJ;

```




EXEMPLES NUMERIQUES ET COMPARAISON DES METHODES

I. ETUDE THEORIQUE DES EXEMPLES.

1.1. Définitions.

Soit $H^q = H^q[a,b]$ l'espace de Hilbert des fonctions réelles définies sur $[a,b]$, ayant une dérivée $q-1$ ième continue et dont la dérivée q ième est de carré sommable. Le produit scalaire sur H^q est défini par :

$$(f | g)_q = \sum_{i=0}^q \int_a^b f^{(i)}(x) g^{(i)}(x) dx$$

(on a $\| f \|_q^2 = (f | f)_q$).

Soit $H^0 = H^0[a,b]$ l'espace de Hilbert des fonctions réelles définies sur $[a,b]$, de carré sommable. H^0 est muni du produit scalaire :

$$(f | g)_0 = \int_a^b f(x) g(x) dx \quad (\| f \|_0^2 = (f | f)_0).$$

D^q désigne l'application linéaire et continue de H^q sur H^0 qui, à $f \in H^q$, fait correspondre $f^{(q)} \in H^0$.

Soit $x \in [a, b]$. Si $q \geq 1$, la fonctionnelle : $f \in H^0 \rightarrow f(x)$ est linéaire et continue et on peut trouver $k \in H^q$ tel que : $(k | f)_q = f(x)$.

1.2. Exemples. (cf Atteia [3]).

On pose : $X = H^q[a, b]$, $Y = H^0[a, b]$, $T = D^q$. Le noyau de T est l'ensemble $\mathcal{N}(T) = \pi^{q-1}$ des polynômes de degré $q-1$. ($\dim \mathcal{N}(T) = q$).

Soit $a \leq x_1 < x_2, \dots < x_n \leq b$. ($n > q$). A chaque x_i , $i \in \{1, \dots, n\}$ on fait correspondre $k_i \in H^q$ tel que $(k_i | f)_q = f(x_i)$ pour tout $f \in H^q$.

$$\text{Soit } L = \{-k_1, \dots, -k_n\} \cup \{k_1, \dots, k_n\}$$

$$\text{et } C = \{f \in H^q \mid \alpha_i \leq f(x_i) \leq \beta_i\} \neq \emptyset.$$

L'ensemble L vérifie les hypothèses (H1) et (H4). On suppose que C vérifie les hypothèses (H2) et (H3).

Le problème (P) peut alors s'écrire :

" trouver $\sigma \in C$ vérifiant :

$$\int_a^b [\sigma^{(q)}(x)]^2 dx = \min \left(\int_a^b [f^{(q)}(x)]^2 dx \mid f \in C \right)$$

Le théorème d'unicité (prop. 1.8) entraîne que le problème (P) a une solution unique σ . Si on pose $K_\sigma = \{f \in H^q \mid f(x_i) = \sigma(x_i) \quad i = 1, \dots, n\}$ alors σ est la fonction spline relative à K_σ : σ est formée de morceaux de polynômes de degré $2q - 1$ qui se raccordent aux points d'abscisses x_i , $i = 1, \dots, n$, jusqu'à la dérivée $2q-2$ ^{ième} comprise (cf Schönberg [23]). Le théorème de caractérisation (prop. 1.2) nous permet de préciser les propriétés de σ :

" une condition nécessaire et suffisante pour que σ soit solution est que l'on ait

$$T^* T \sigma = \sum_{i=1}^n \lambda_i k_i,$$

$$\lambda_i \geq 0 \quad \text{si } \sigma(x_i) = \alpha_i,$$

$$\lambda_i \leq 0 \quad \text{si } \sigma(x_i) = \beta_i,$$

$$\lambda_i = 0 \quad \text{si } \sigma(x_i) \in]\alpha_i, \beta_i[."$$

Alors $T^* T\sigma \in \text{Range}(T^*) = \mathcal{J}(T)$ donc :

$$\sum_{i=1}^n \lambda_i f(x_i) = (T^* T\sigma | f)_q = 0 \quad \forall f \in \Pi^{q-1}$$

Soit $f \in H^q$. La formule de Taylor (avec reste sous forme intégrale) donnant :

$$f(x) = (f(a) + (x-a)f'(a) + \dots + \frac{(x-a)^{q-1}}{(q-1)!} f^{(q-1)}(a) + \int_a^b \frac{(x-t)^{q-1}}{(q-1)!} f^{(q)}(t) dt)$$

$$(u_+ = \begin{cases} u & \text{si } u > 0 \\ 0 & \text{si } u \leq 0 \end{cases}),$$

on a alors :

$$\begin{aligned} (T^* T\sigma | f)_q &= \int_a^b \sum_{i=1}^n \lambda_i \frac{(x_i - t)^{q-1}}{(q-1)!} f^{(q)}(t) dt \\ &= (\varphi | Tf)_0. \end{aligned}$$

Les propriétés des fonctions "spline" d'interpolation mises en évidence par Laurent et Anselone [1] permettent de conclure : $T(\sigma) = \sigma^{(q)} = \varphi$. Pour tout $i \in \{1, \dots, n\}$, le nombre $\lambda_i (-1)^q$ représente le saut de dérivée $2q - 1$ ^{ième} de σ au point d'abscisse x_i :

$$(7.1) \quad \underline{\sigma^{(2q-1)}(x_i^+) - \sigma^{(2q-1)}(x_i^-) = (-1)^q \lambda_i.}$$

Remarque :

Si l'hypothèse (H3) n'est pas vérifiée, C est défini par des conditions d'interpolation. C vérifie alors la propriété (H3') et la proposition 1.5 nous donne alors :

"une condition nécessaire et suffisante pour que σ soit solution est que l'on ait :

$$T^* T\sigma = \sum_{i \in I_1} \lambda_i k_i + \sum_{i \in I_2} \mu_i k_i$$

$$\lambda_i \leq 0 \quad \text{si} \quad \sigma(x_i) = \beta_i$$

$$\lambda_i \geq 0 \quad \text{si} \quad \sigma(x_i) = \alpha_i$$

$$\lambda_i = 0 \quad \text{si} \quad \sigma(x_i) \in]\alpha_i, \beta_i[."$$

1.3. Mise en oeuvre des méthodes directes.

A chaque itération ν , on détermine une direction de descente s^ν et une telle direction est toujours définie à partir de Ωy^ν .

Soit σ_y^ν la fonction spline associée au point y^ν . D'après la remarque faite au paragraphe 2.4 du chapitre 2 et la propriété (7-1), le vecteur Ωy^ν a pour $i^{\text{ème}}$ composante le saut de dérivée $2q-1^{\text{ème}}$ de σ_y^ν au point x_i . Etant donné y^ν il est aisé de calculer les sauts de dérivée $2q-1^{\text{ème}}$ de Ωy^ν : nous donnons en annexe différentes procédures permettant d'effectuer ces calculs (procédures SPLUN, SPLDEU, SAUDERIV) (cf Carasso [6], Carasso et Laurent [7]).

Dans les méthodes de relaxation, on a besoin de connaître la matrice Ω elle-même. On peut la calculer à partir des fonctions "spline" de base: nous avons vu, au paragraphe 3.3 du chapitre II, que pour tout $y \in \mathbb{R}^n$, $\sigma_y = \sum_{i=1}^n y_i \sigma_i$. Par conséquent :

$$\sigma_y^{(2q-1)}(x_j^+) - \sigma_y^{(2q-1)}(x_j^-) = \sum_{i=1}^n y_i [\sigma_i^{(2q-1)}(x_j^+) - \sigma_i^{(2q-1)}(x_j^-)]$$

$$\text{donc } \Omega_{ij} = \sigma_i^{(2q-1)}(x_j^+) - \sigma_i^{(2q-1)}(x_j^-).$$

et les éléments de la matrice Ω sont les sauts de dérivée $2q-1^{\text{ème}}$ des fonctions "spline" de base.

II - EXEMPLES NUMERIQUES - $q, n.$

$$x_i = (i-1)p + x_0 \quad i = 1, \dots, n \quad (p = \text{pas})$$

f , fonction définie sur $[x_0, (n-1)p + x_0]$. ε , nombre positif.

$$C = \{g \in H^q \mid f(x_i) - \varepsilon \leq g(x_i) \leq f(x_i) + \varepsilon, i=1, \dots, n\}$$

$$P = \{y \in R^n \mid f(x_i) - \varepsilon \leq y_i \leq f(x_i) + \varepsilon, i=1, \dots, n\}.$$

Etant donné $y \in P$ défini par l'une quelconque des méthodes, on détermine un

"candidat à la solution" $\bar{\sigma} = \sigma_{\bar{y}}$. On utilise pour ceci le théorème de caractérisation (proposition 3.1) qui nous permet de "déterminer" $I_+(\bar{y})$ et $I_-(\bar{y})$. $\bar{\sigma}$ est la fonction "spline" relative à :

$$\bar{C} = \{g \in H^q \mid g(x_i) = \begin{cases} f(x_i) + \varepsilon & \text{si } i \in I_+(\bar{y}) \\ f(x_i) - \varepsilon & \text{si } i \in I_-(\bar{y}) \end{cases}.$$

si $\bar{\sigma} \in C$ et si les sauts de dérivée $2q-1^{\text{ème}}$ de $\bar{\sigma}$ sont des signes convenables, alors $\bar{\sigma}$ est solution.

C'est selon ce principe qu'ont été vérifiés les résultats que nous donnons ici sous forme de courbes ou de tableaux. (Les procédures utilisées pour vérifier ces résultats sont données en Annexe).

Tableaux représentant les sauts de dérivées des fonctions "spline" en fonction de l'ordre :

$$f(x) = \frac{2x^2 + x - 1}{x^2 - x + 1} \quad n = 10 \quad x_0 = 0 \quad p = 1 \quad \varepsilon = 0,8$$

i	1	2	3	4	5	6	7	8	9	10
q = 1	+				-	+	+			
q = 2			+		-	+				-
q = 3	-	+	-	+	-	+	-	+	+	-

$$f(x) = \frac{2x^2 + x - 1}{x^2 - x + 1} \quad n = 5 \quad x_0 = -4 \quad p = 1 \quad \xi = 0,8$$

i	1	2	3	4	5
q = 1	+		-	+	
q = 2		+	-	+	-
q = 3	-	+	-	+	-

Les signes + et - indiquent les signes des sauts des dérivées $2q-1^{\text{ème}}$ (multipliés par $(-1)^q$) de la fonction "spline". D'après le théorème de caractérisation et la propriété (7 -1) de tels tableaux représentent sans ambiguïté les fonctions "spline".

(voir aussi les courbes données à la fin de ce chapitre).

III-COMPARAISON DES METHODES-

Présentation des résultats :

Le nombre appelé erreur relative et mentionné dans les tableaux est défini comme suit :

$$\rho(y^v) = \frac{d - y^v \tau_{\Omega y^v}}{d}$$

La valeur théorique d est calculée à partir de la solution théorique y_0 que l'on connaît dès que l'on a déterminé σ (cf paragraphe précédent).

Pour tous les exemples numériques que nous allons présenter nous convenons de prendre comme point de départ le point

$$y^0 = [f(x_1), \dots, f(x_n)].$$

Les temps machine qui seront donnés sont les temps d'exécution sur IBM 360-40 : ils sont donnés en secondes.

3.1. Comparaison des procédures GRADPROJ1, GAUSS-SEIDEL, SOUTHWELL ET FRANCKWOLFE

GRADPROJ1	algorithme n° 1
SOUTHWELL	" n° 2
GAUSS-SEIDEL	" n° 3
FRANCK-WOLFE	" n° 4 (cas particulier-cf chapitre V)

Etude dans le cas où $f(x) = \sin x$ ($x_0 = 0, p=1, n=10, q=2$)

Nombre d'itérations

ϵ	0.1	0.3	0.5
GRADPROJ1	9(3)	11(1)	23(3)
SOUTHWELL	2	3	6
GAUSS-SEIDEL	4	8	14
FRANCKWOLFE	>34($2 \cdot 10^{-3}$)	>47($5 \cdot 10^{-2}$)	>117($2 \cdot 10^{-1}$)
erreur relative	10^{-3}	10^{-2}	$2 \cdot 10^{-2}$
valeurs theorique	3.212942	1.28353	0.33702

Temps d'exécution correspondants

ϵ	0.1	0.3	0.5
GRADPROJ1	8,7	9,4	19,9
SOUTHWELL	0,9	1,3	2,6
GAUSS-SEIDEL	1,6	3,2	5,6
FRANCKWOLFE	>30	>42	>103

Etude dans le cas où $f = \frac{2x^2+x-1}{x^2-x+1}$ ($x_0 = -4, p=1, n=10, q=2$)

Nombre d'itérations

ϵ	0.2	0.4	0.6	0.8
GRADPRØJ1	14	34(1)	57(2)	90(2)
GAUSS-SEIDEL	4	8	18	23
SOUTHWELL	2	6	9	8
FRANCKWØLFE	$>14(5 \cdot 10^{-3})$	$>27(2 \cdot 10^{-2})$	$>58(4 \cdot 10^{-2})$	$93(9 \cdot 10^{-2})$
erreur relative	$5 \cdot 10^{-4}$	10^{-3}	10^{-3}	$5 \cdot 10^{-3}$
valeur théorique	24.08709	12.82923	6.11255	2.86155

Temps d'exécution correspondants.

ϵ	0.2	0.4	0.6	0.8
GRADPRØJ1	11,5	28,1	47,7	75
GAUSS-SEIDEL	1,6	3,2	5,4	6,9
SOUTHWELL	0,9	2,7	3,9	3,5
FRANCKWØLFE	>13	>24	>52	>83

Remarques :

Pour la procédure GRADPRØJ1, le chiffre entre parenthèses indique le nombre de fois où l'on a utilisé la direction de descente modifiée (cf III-paragraphe 2). Dans le premier cas on avait $\Delta = E/20$ dans le deuxième $\Delta = E/50$.

Pour la procédure FRANCKWØLFE, le chiffre entre parenthèses indique la précision obtenue pour le nombre d'itération donné.

Pour les procédures utilisant les méthodes de relaxation, on appelle itération ce que l'on obtient après n relaxations.

La procédure GRAD utilisée ici est la procédure SPLDEU: dix appels de cette procédure demandent 3,6s.

Efficacité de ces méthodes.

Les méthodes de relaxation sont les plus efficaces : la procédure SOUTHWELL puis la procédure GAUSS-SEIDEL (l'aspect automatique de certaines opérations réduit considérablement les temps de calcul). Quant à la procédure FRANCK WOLFE, elle conduit à des calculs trop longs pour être utilisée pour le calcul des fonctions "spline".

3.2 Procédure FRANCØNJ.

FRANCØNJ algorithme n° 4 et 4'

Etude dans le cas où $f(x) = \sin x$. ($x_0=0$, $p=1$, $n=10$, $q=2$).

Nous allons comparer les résultats numériques obtenus en faisant varier les paramètres R et L1 de FRANCØNJ.

cas 1 : R = EP×2, L1 à la valeur "faux", RØ peut prendre n'importe quelle valeur (algorithme de Franck et Wolfe).

cas 2 : R = EP/2, L1 a la valeur "vrai", RØ = EP×2 (alg. n° 4')

cas 3 : R = EP/5, L1 a la valeur "vrai", RØ = EP×2 (alg. n° 4')

cas 4 : R = EP/5, L1 a la valeur "faux", RØ quelconque (alg n° 4).

Nombre d'itération

ϵ	0.1	0.3			0.5		
erreur relative	10^{-3}	10^{-2}	$5 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	10^{-1}	$6 \cdot 10^{-2}$	$2 \cdot 10^{-2}$
cas 1	> 34	> 47			> 117		
cas 2	4	59	> 71		69	> 75	
cas 3	4	8	9	39	12	15*	74
cas 4	4	12	23	62	19	21	78

ϵ	0.1	0.3	0.5
erreur relative	10^{-3}	$5 \cdot 10^{-3}$	$6 \cdot 10^{-2}$
cas 1	>42		
cas 2	4,9	> 88	>93
cas 3	5,3	13,3	23,2
cas 4	4,9	28,3	25,8

Efficacité :

Le choix de la direction de descente est d'autant plus efficace que les voisinages K^v sont plus "petits" devant P. D'autre part l'"amélioration" de la dissection de descente a des effets variables selon les exemples choisis. La recherche d'une direction de descente "meilleure" donne des résultats satisfaisants pour les premières itérations (cf temps ci dessus). Pour des erreurs relatives assez faibles le temps utilisé par la recherche d'une amélioration n'est pas compensé par cette amélioration. On peut aussi remarquer que pour des erreurs relatives assez fortes, on peut comparer la procédure FRANCØNJ dans le cas 4 avec la procédure GRADPROJ2. Mais la procédure GRADPROJ2 est meilleure pour des erreurs relatives plus faibles.

3.3 Procédure GRADCØNJSIMPLE

Nous allons comparer les procédures GRADCØNJSIMPL et GRADPROJ1 dans le cas où

$$f(x) = \frac{2x^2 + x - 1}{x^2 - x + 1} \quad \text{avec } x_0 = -4, p=1 \text{ ou } 0.75$$

$$\text{DELTA} = \text{EP}/50 \quad R\emptyset = 10^{-6} \quad T\emptyset = 5 \cdot 10^{-5}$$

Nombre d'itération

ϵ	0.2	0.4	0.6	0.8	0.2	0.4
GRADPRØJ1	14	8	24(1)	11	9	23(3)
GRADCØNJSIMPLE	24	13	24	23	10	51
erreur relative	10^{-3}	$3 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$2 \cdot 10^{-1}$	$7 \cdot 10^{-3}$	$3 \cdot 10^{-3}$
	n = 10				n = 15	

Temps d'exécution correspondants (n=10)

ϵ	0.2	0.4	0.6	0.8
GRADPRØJ1	11,5	6,7	20,6	9
GRAOCØNJSIMPL	>20	>15	>25	>20

Pour les erreurs relatives assez fortes, la procédure GRADCØNJSIMPL est moins bonne que la procédure GRADPRØJ1 et plus on exige une erreur relative faible, plus GRADPROJ est mauvaise. On a en outre des difficultés dues à la structure de la procédure GRADCØNJ : il est difficile de bien ajuster les tests ; d'autre part il faut éviter de tourner en rond à cause des "direction conjuguées".

3.4 Difficultés propres à certains exemples numériques

Les tableaux suivants indiquent l'importance des paramètres n et p pour la rapidité de l'obtention du résultat avec une procédure donnée : plus les contraintes sur les abscisses sont "lâches" ou plus les contraintes sur les "bâtonnets" sont "sévères",

plus la convergence d'une méthode, quelle qu'elle soit, est rapide.

Notation :

ν indique le nombre d'itérations nécessaire avec la procédure GRADPROJ pour avoir une erreur relative de 10^{-3} seulement.

$$f(x) = \sin x.$$

ϵ	ν
0.1	42
0.3	425
0.5	>750

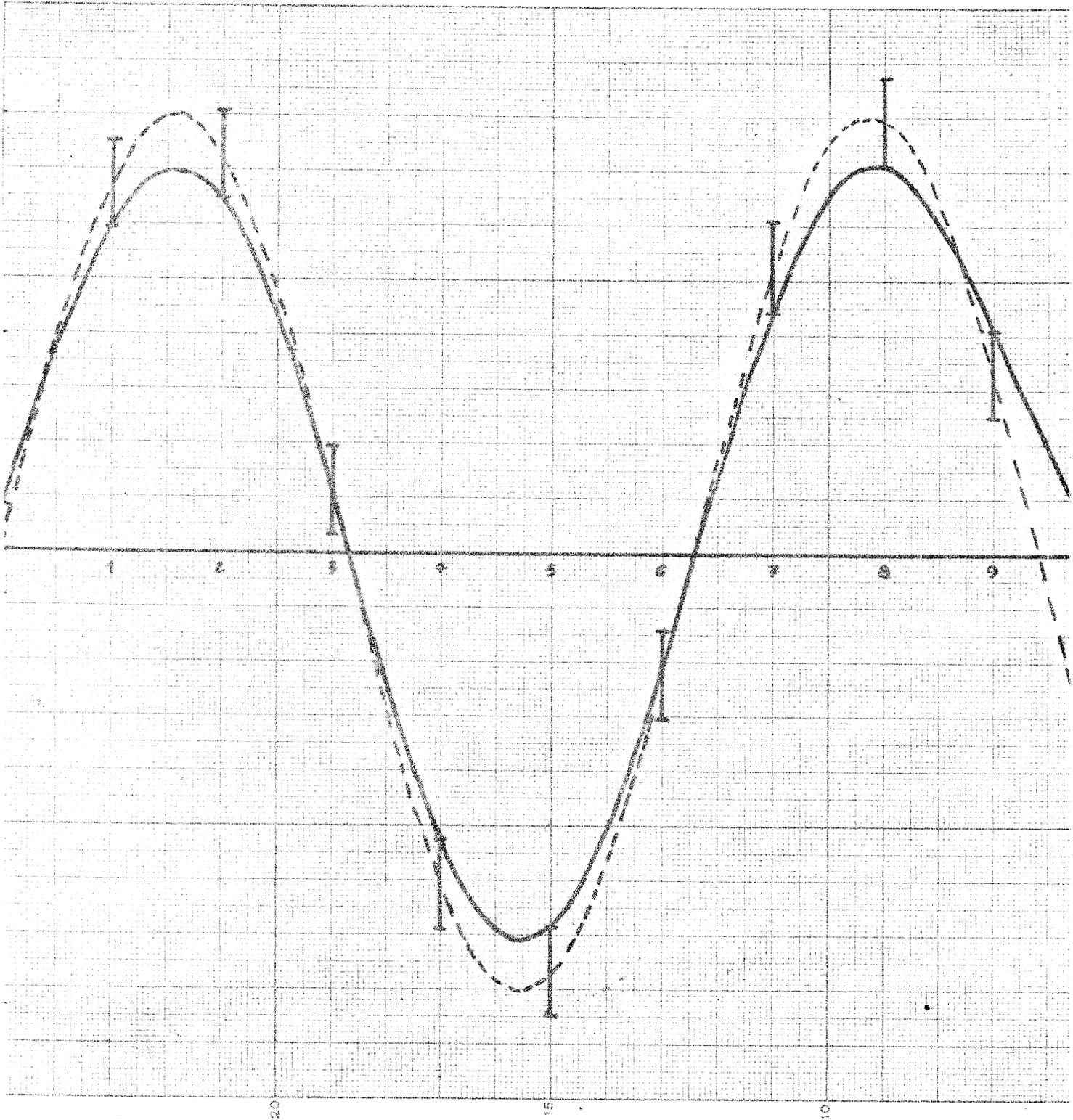
1) $n = 10$
 $p = 0.5$

2) $\epsilon = 0.1$

n	10	10	15	15	20	20	30	40
ρ	0,5	1	0,375	0.75	0.25	0.5	0.75	0.75
ν	42	9	>130	12	>941	30	14	24

$$f(x) = \frac{2x^2 + x - 1}{x^2 - x + 1}$$

n	p	ν 0.2	ν 0.4
5	2	6	6
10	1	14	33
15	0.75	35	>79($3 \cdot 10^{-3}$)
20	0.5	>65(10^{-2})	290



$$n = 10$$

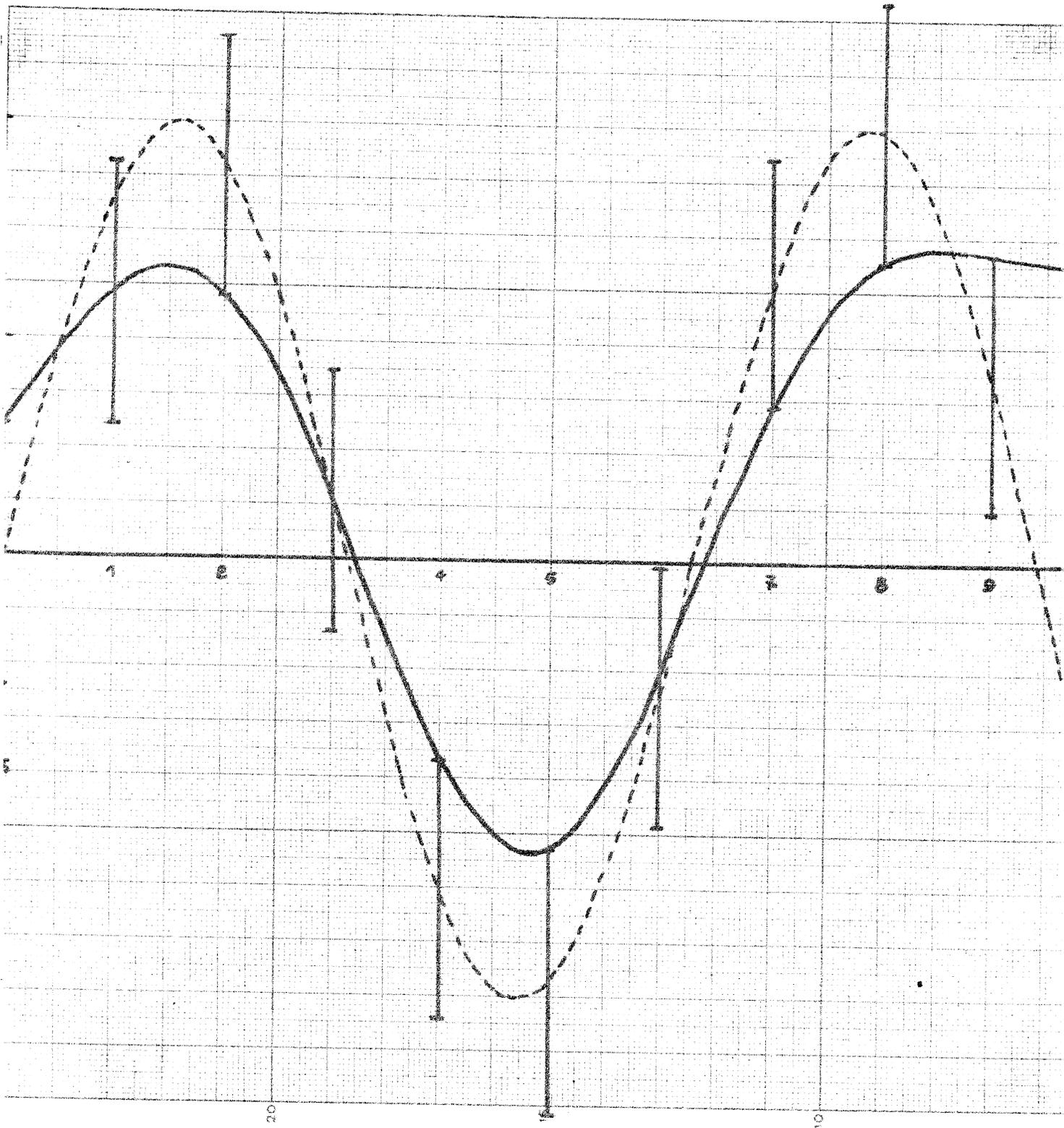
$$q = 2$$

$$y(x) = \sin(x)$$

$$x_i = i - 1$$

$$y(x_i) \in [\sin(x_i) - 0.1, \sin(x_i) + 0.1]$$

$$i = 1, \dots, n.$$



$$n = 10$$

$$q = 2$$

$$y(x) = \sin(x)$$

$$x_i = i - 1$$

$$y(x_i) \in [\sin(x_i) - 0.3, \sin(x_i) + 0.3]$$

$$i = 1, \dots, n.$$



$n = 10$

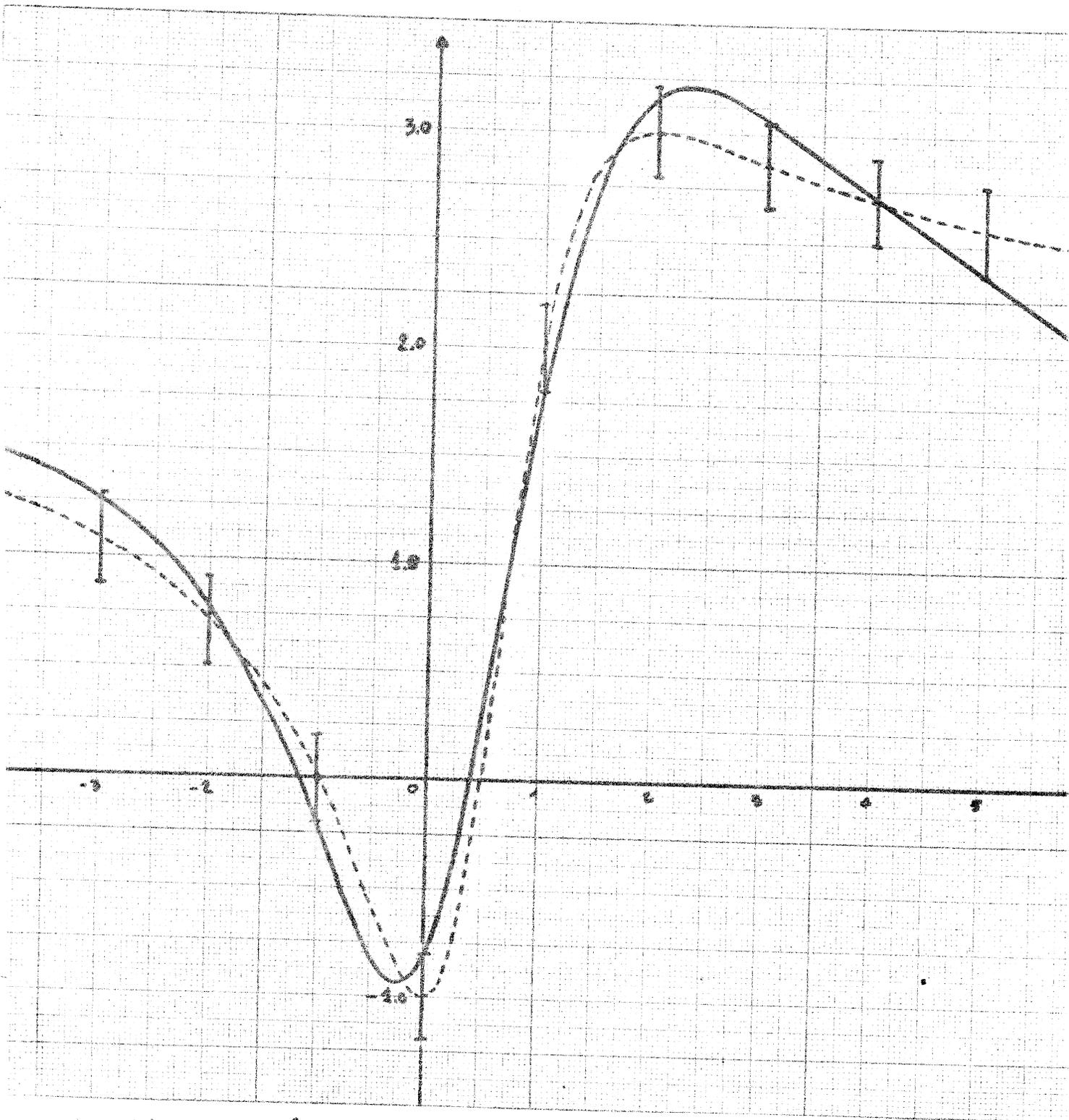
$q = 2$

$y(x) = 5 \sin(x)$

$x_i = i - 1$

$y(x_i) \in [\sin(x_i) - 0.5, \sin(x_i) + 0.5]$

$i = 1, \dots, n$

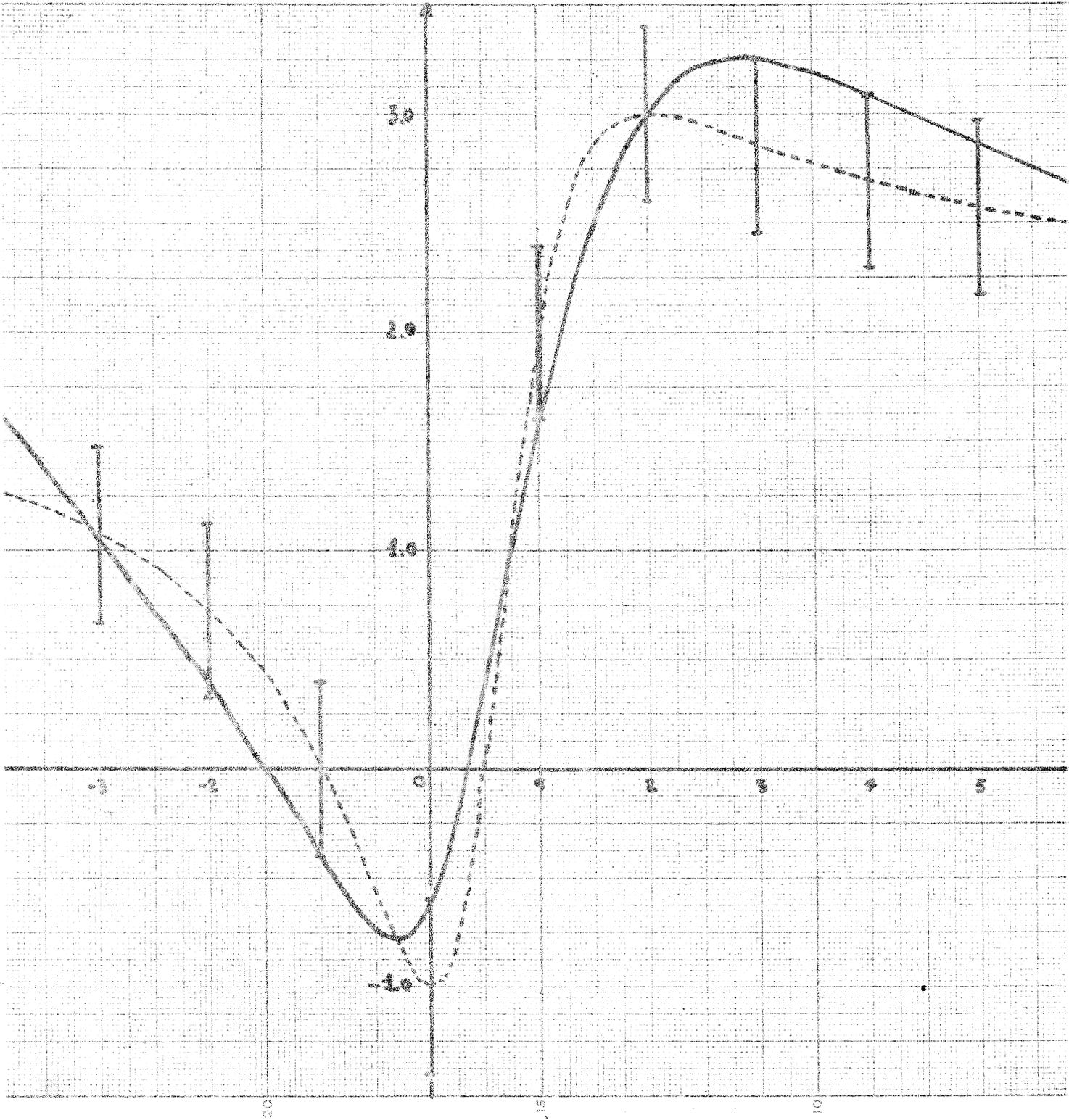


$$n = 10, \quad q = 2$$

$$f(x) = \frac{2x^2 + x - 1}{x^2 - x + 1}$$

$$x_i = -4 + (i-1) \quad i = 1, \dots, n$$

$$\pm 0.2$$

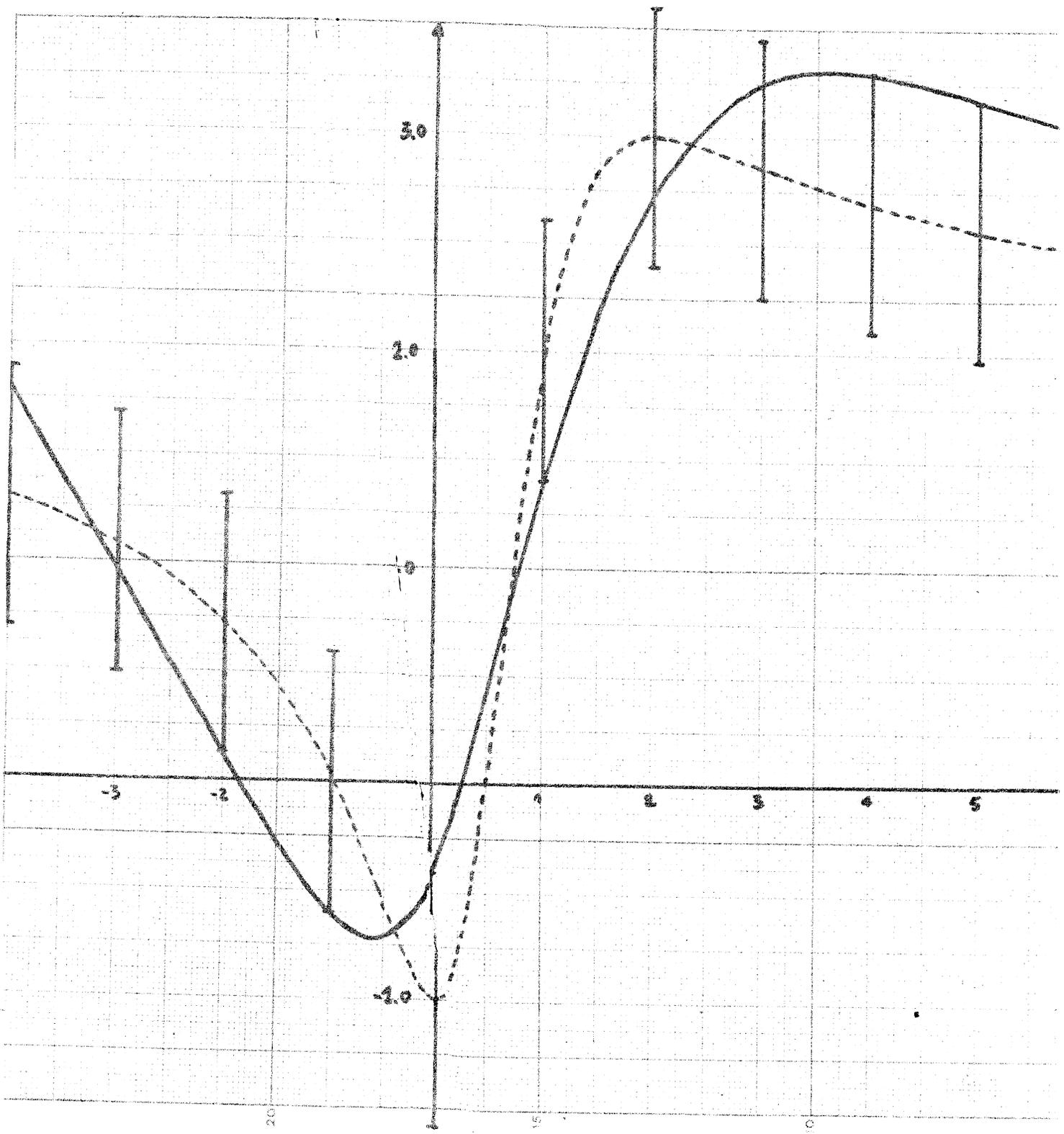


$n = 10, \quad q = 2$

$x_i = -4 + (i-1) \quad i = 1, \dots, n.$

$$y(x) = \frac{2x^2 + x - 1}{x^2 - x + 1}$$

± 0.4

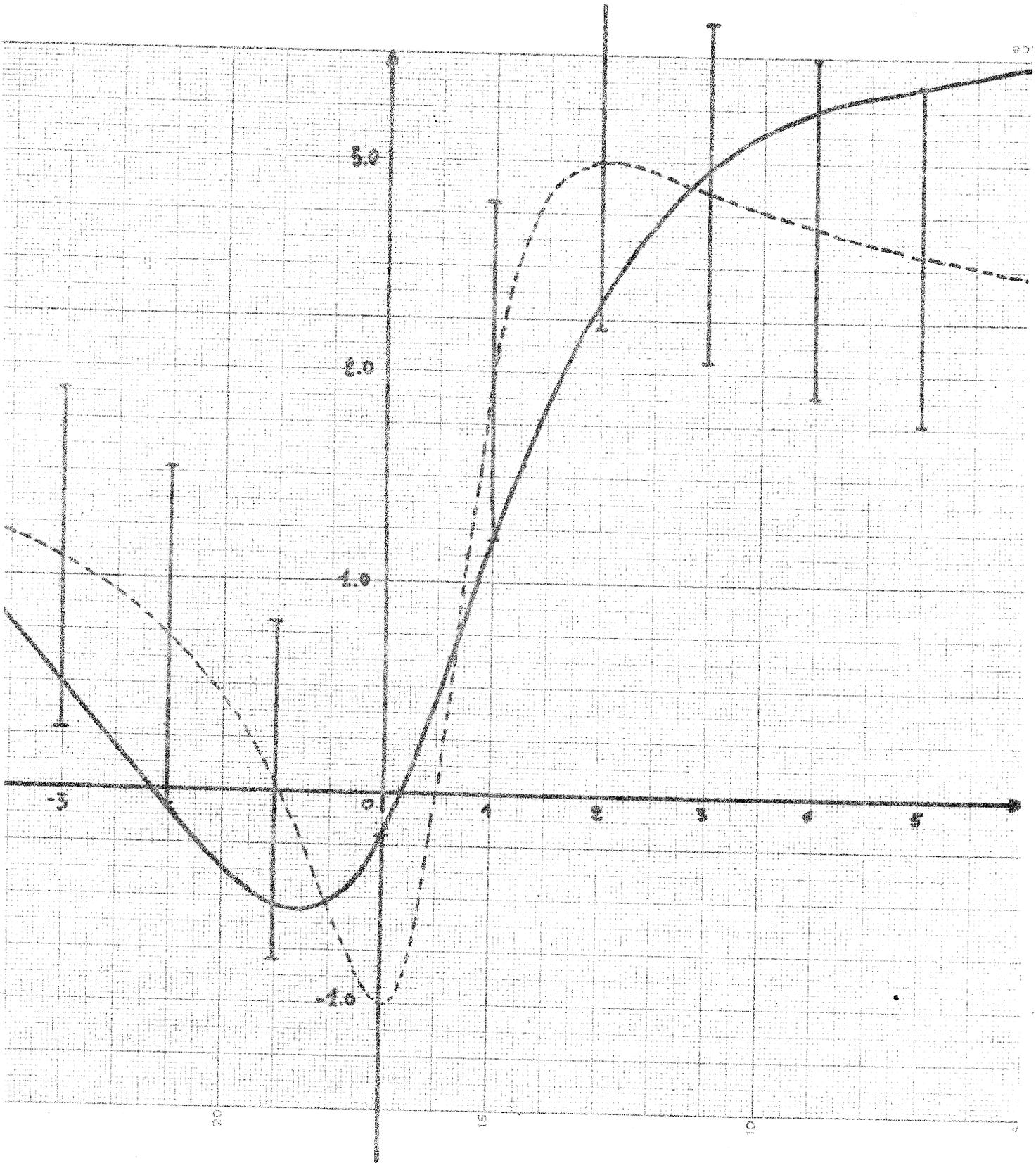


$n = 10, \quad q = 2$

$$y(x) = \frac{2x^2 + x - 1}{x^2 - x + 1}$$

$$x_i = -4 + (i-1) \quad i = 1, \dots, n$$

$$\pm 0.6$$



$n = 10, q = 2$

$y(x) = 2x^2 + x - 1$
 $x^2 - 2 + 1$

$x_i = -4 + (i-1) \quad (i=1, \dots, n)$

± 0.8

D E U X I E M E

P A R T I E



M E T H O D E S D U A L E S

C H A P I T R E V I I I



ETUDE GEOMETRIQUE DU PROBLEME DES FONCTIONS

" SPLINE " DANS UN CONVEXE .

Ce chapitre est consacré à une étude géométrique du problème (P) : dans un premier paragraphe, nous étudions les propriétés d'une famille particulière de sous-ensembles de L et cette famille nous permet de donner une caractérisation simple du convexe $T(C)$ au paragraphe 2. Dans un troisième paragraphe, nous définissons le problème dual associé au problème (P) : l'algorithme qui sera proposé au chapitre suivant résoud ce problème dual et il est fondé sur la caractérisation de $T(C)$ donnée au paragraphe 2.

I-PROPRIETES DE SOUS ENSEMBLES PARTICULIERS DE L.-

Notations et définitions.

Soit $L = \{l_1, \dots, l_m\}$. On pose $l_i = l_i^1 + l_i^2$ avec $l_i^1 \in \mathcal{R}(\Gamma)$ et $l_i^2 \in \mathcal{R}(T)^\perp$ pour tout $i \in \{1, \dots, m\}$.

$$C = \{x \in X \mid (l_i \mid x)_X \leq c_i, \quad i = 1, \dots, m\} \neq \emptyset .$$

$$\underline{Q} = T(C)$$

Supposons que L vérifie les hypothèses (H1) et (H4). Nous allons tout d'abord montrer deux lemmes dont nous aurons souvent besoin par la suite.

Lemme 8.1 " Soit $\{ l_{i_1}, \dots, l_{i_r} \} \subset L$ tels que $l_{i_j} \neq \pm l_{i_k}$ pour $j \neq k$. Un élément de la forme :

$$l = \sum_{j=1}^r \alpha_j l_{i_j}, \quad r \leq q+1, \quad \underline{l \neq 0 \text{ (avec des } \alpha_j \text{ non tous nuls)}}$$

ne peut appartenir à $\mathcal{N}(T)^\perp$ que si $r = q+1$ ".

Démonstration :

Supposons $r = q$. (On "complète" éventuellement avec des α_j nuls). Comme $l \in \mathcal{N}(T)^\perp$ le système linéaire homogène :

$$\sum_{j=1}^q \alpha_j (l_{i_j} | \omega_k)_X = 0 \quad \forall k \in \{1, \dots, q\}$$

a une solution non nulle. Donc $\det [(l_{i_j} | \omega_k)_X] = 0$ ce qui contredit l'hypothèse (H4). Donc $r = q+1$.

Lemme 8.2 " Soit $\{ l_{i_1}, \dots, l_{i_{q+1}} \} \subset L$ tels que $l_{i_j} \neq \pm l_{i_k}$ pour $j \neq k$. Alors les éléments non nuls de $\mathcal{N}(T)^\perp$, de la forme $\sum_{j=1}^{q+1} \alpha_j l_{i_j}$, sont tous proportionnels "

Démonstration :

L'hypothèse (H4) entraîne que le système

$$\sum_{j=1}^q \alpha_j (l_{i_j} | \omega)_X = -\alpha_{q+1} (l_{i_{q+1}} | \omega_k)_X \quad \forall k \in \{1, \dots, q\}$$

a une solution unique pour chaque α_{q+1} . Celle-ci dépend donc linéairement de α_{q+1} , d'où le lemme.

Définition

Soit $J \subset \{1, \dots, m\}$. On dit que J appartient à l'ensemble \mathcal{J} si J vérifie les propriétés suivantes :

(8.1) J contient exactement $q+1$ indices,

(8.2) $l_i \neq \pm l_j$ si $i, j \in J$ $i \neq j$

(8.3) Il existe $q+1$ nombres positifs ou nuls λ_j^J , $j \in J$, tels que :

$$\sum_{j \in J} \lambda_j^J = 1 \quad \text{et} \quad l_J = \sum_{j \in J} \lambda_j^J l_j \in \mathcal{P}(T)^\perp.$$

Lemme 8.3 " Si les hypothèses (H2) et (H3) sont vérifiées, alors \mathcal{J} n'est pas vide; pour tout $J \in \mathcal{J}$, l_J est unique et pour tout $j \in J$, nous avons $\lambda_j^J > 0$ ".

Démonstration :

L'hypothèse (H2) entraîne qu'il existe une fonction "spline" σ relative à C et à T telle que $T\sigma \neq 0$. Si on pose :

$$I_\sigma = \{ i \in [1, \dots] \mid (l_i | \sigma)_X = \alpha_i \}$$

la proposition 1.2 (théorème de caractérisation) nous donn alors :

$$T^*T\sigma = - \sum_{i \in I_\sigma} \lambda_i l_i,$$

où les λ_i sont des nombres positifs ou nuls, non tous nuls, et les hypothèses (H1) et (H3) entraînent :

$$l_j \neq \pm l_k \quad k, j \in I_\sigma, \quad k \neq j.$$

$$\sum_{i \in I_\sigma} \lambda_i l_i \in \text{Range}(T^*) = \mathcal{P}(T)^\perp \quad \text{donc} \quad \sum_{i \in I_\sigma} \lambda_i l_i^1 = 0$$

(T) étant de dimension q , le théorème de Carathéodory (Eggleston [10]) nous dit qu'il existe $\{i_1, \dots, i_r\} \subset I_0$, avec $r \leq q+1$, tels que :

$$0 = \sum_{j=1}^r \mu_{i_j} l_{i_j}^1 \quad \text{avec} \quad \sum_{j=1}^r \mu_{i_j} = 1 \quad \text{et} \quad \mu_{i_j} \geq 0 \quad j = 1, \dots, r$$

Posons alors $J = \{i_1, \dots, i_r\}$.

Comme $\sum_{j=1}^r \mu_{i_j} l_{i_j}^1 \in \mathcal{P}(T)^+$, le lemme 7.1 entraîne $r = q+1$ et $\mu_j > 0$ pour tout

$j \in J$. Par conséquent $J \in \mathcal{J}$.

De plus l'égalité $\sum_{j \in J} \mu_j = 1$ et le lemme 7.2 entraînent que $l_J = \sum_{j \in J} \mu_j l_j$ est

l'unique combinaison convexe d'éléments de J .

Proposition 8.2

"Soit l un élément de X , combinaison linéaire, à coefficients positifs, d'éléments distincts et non opposés de L . Si $l \in \mathcal{P}(T)^+$, alors l est combinaison linéaire à coefficients positifs, des $l_j, J \in \mathcal{J}$ ".

Démonstration :

Soit $l \in \mathcal{P}(T)^+$, $l = \sum_{i \in I} \lambda_i l_i$ avec $I \subset \{1, \dots, m\}$, $\lambda_i > 0$ pour tout $i \in I$,

$l_i \neq \pm l_j$ si $i \neq j$, i et $j \in I$.

Le lemme 8.1 entraîne que I contient au moins $q+1$ éléments.

D'autre part, comme $l^1 = \sum_{i \in I} \lambda_i l_i^1 = 0$, d'après le théorème de Carathéodory

on peut trouver $J \subset I$, J contenant $q+1$ indices, d'une part et $q+1$ nombres positifs ou nuls μ_j , $j \in J$, d'autre part, tels que :

$$\sum_{j \in J} \mu_j l_j^1 = 0 \quad \text{et} \quad \sum_{j \in J} \mu_j = 1$$

Alors $J \in \mathcal{J}$ et $l_J = \sum_{j \in J} \mu_j l_j$ avec $\mu_j > 0$ pour tout $j \in J$.

Soit $l_1 = l - \theta_J l_J = \sum_{i \in I} \lambda_i l_i - \theta_J \sum_{j \in J} \mu_j l_j$,

$$\text{avec } \theta_J = \min \left(\frac{\lambda_i}{\mu_i} \mid i \in J \right) > 0$$

Posons $I_1 = \left\{ i \in I \mid \theta_J = \frac{\lambda_i}{\mu_i} \right\}$.

Alors $l = l_1 + \theta_J l_J$

$$\text{avec } l_1 = \sum_{i \in I - I_1} \lambda_i l_i \text{ et } \lambda_i = \begin{cases} \lambda_i > 0 & \text{si } i \notin J \\ \lambda_i - \theta_J \mu_i > 0 & \text{si } i \in J \end{cases}$$

On voit donc qu'après $n-q-1$ "décompositions", au plus, on peut mettre l sous la forme :

$$l = \sum_{J \in \mathcal{J}} \mu_J l_J \text{ avec } \mu_J \geq 0, J \in \mathcal{J}.$$

II-CARACTERISATION DE $T(C) = \Omega$

Nous supposons que les propriétés (H1), (H2), (H3) et (H4) sont vérifiées.

Notations

$J = \{i_1, \dots, i_{q+1}\} \in \mathcal{J}$. $l_J = \sum_{j \in J} \lambda_j^J l_j \in \mathcal{P}(T)^\perp = \text{Range}(T^*)$.

Posons :

$$C_J = \{x \in X \mid (l_j \mid x)_X \leq c_j, j \in J\}$$

$$\Omega_J = \{y \in Y \mid (\varphi_J \mid y)_Y \leq \gamma_J\}$$

où $\varphi_J = T^{*-1}(l_J)$ et $\gamma_J = \sum_{j \in J} \lambda_j^J c_j$

Lemme 8.4 :
$$C = \bigcap_{J \in \mathcal{Y}} C_J$$

Démonstration

Nous allons montrer que pour tout $i \in \{1, \dots, m\}$, il existe $J_i \in \mathcal{Y}$ tel que $i \in$

Le lemme se déduit alors immédiatement de la définition de C_J , $J \in \mathcal{Y}$.

Soit $i \in \{1, \dots, m\}$ et $J \in \mathcal{Y}$: un tel J existe bien, d'après le lemme 8.3

On peut toujours trouver $i_0 \in J$ tel que $I = J + \{i\} - \{i_0\}$ vérifie la propriété (8.2). L'hypothèse (H4) et le lemme 8.1 entraînent alors qu'il existe $q+1$ nombres non nuls μ_j , $j \in I$, tels que :

$$l_0 = \sum_{j \in I} \mu_j l_j \in \mathcal{R}(T)^\perp.$$

Posons $\mu_{i_0} = 0$ et considérons :

$$l = \sum_{j \in I + \{i_0\}} \mu_j l_j - \theta_J \sum_{j \in J} \lambda_j^J l_j = l_0 - \theta_J l_J$$

$$\text{avec } \theta_J = \frac{\mu_{i_1}}{\lambda_{i_1}^J} = \min \left(\frac{\mu_j}{\lambda_j^J} \mid j \in J \right)$$

On pose :

$$\begin{cases} \rho_j = \mu_j - \theta_J \lambda_j^J & \text{si } j \neq i \\ \rho_i = \mu_i \end{cases}$$

et comme $i \in I + \{i_0\} - \{i_1\}$, on peut poser :

$$J_i = I + \{i_0\} - \{i_1\} = J + \{i\} - \{i_0\}$$

Alors $l = \sum_{i \in J_i} \rho_i l_i$ et les propriétés :

$$l \neq 0, l \in \mathcal{R}(T)^\perp, \rho_j \geq 0 \quad \forall j \in J_i,$$

entraînent que J_i vérifie la propriété (8.3); donc, finalement, $J_i \in \mathcal{J}$.

Remarque

En fait à chaque i on ne fait pas correspondre de façon unique J_i : il peut y avoir plusieurs choix pour J_i .

Lemme 8.5 " $T(C_J) = \Omega_J \quad \forall J \in \mathcal{J}$ "

Démonstration :

L'inclusion $T(C_J) \subset \Omega_J$ est évidente : en effet si $x \in C_J$, on a alors :

$$(\varphi_J | Tx)_Y = (\ell_J | x)_X = \sum_{j \in J} \lambda_j^J (\ell_j | x)_X \leq \sum_{j \in J} \lambda_j^J c_j = \gamma_J.$$

Montrons maintenant que $T(C_J) \supset \Omega_J$.

Soit $y \in \Omega_J$. Alors $(\varphi_J | y)_Y \leq \gamma_J$. Soit $i \in J$. Si on pose $\rho_j = c_j$, $j \in J - \{i\}$, et si ρ_i vérifie :

$$\rho_i \lambda_i^J = (\varphi_J | y)_Y - \sum_{j \in J - \{i\}} \lambda_j^J c_j,$$

on a $\rho_i \leq c_i$.

$$\text{Soit } F_{J,\rho}^\perp = \{y \in Y \mid (\varphi_J | y)_Y = \sum_{j \in J} \lambda_j^J \rho_j\}$$

$$\text{et } K_{J,\rho}^\perp = \{x \in X \mid (\ell_j^J | x)_X = \rho_j, j \in J.\}$$

Alors $y \in F_{J,\rho}^\perp$ et on peut montrer (cf Laurent et Anselone [1]) que

$$F_{J,\rho}^\perp = T(K_{J,\rho}^\perp). \text{ En outre, par définition des } \rho_j, K_{J,\rho}^\perp \subset C_J. \text{ Donc } y \in T(C_J).$$

Proposition 8.3.

" $\Omega = \bigcap_{J \in \mathcal{J}} \Omega_J$ "ou encore" $T(\bigcap_{J \in \mathcal{J}} C_J) = \bigcap_{J \in \mathcal{J}} T(C_J)$ "

Démonstration

1) Les lemmes 8.4 et 8.5 entraînent immédiatement l'inclusion :

$$T(C) = T\left(\bigcap_{J \in \mathcal{J}} C_J\right) \subset \bigcap_{J \in \mathcal{J}} T(C_J) = \bigcap_{J \in \mathcal{J}} \Omega_J$$

2) Soit $y \in \bigcap_{J \in \mathcal{J}} \Omega_J$. Supposons que $y \notin \Omega$.

Ω ayant un convexe fermé de Y , il existe un hyperplan fermé qui sépare strictement y et Ω ; ou encore, il existe $\varphi \in Y$, $\varphi \neq 0$ et $\varepsilon > 0$ tels que :

$$(8.4) \quad (\varphi | y)_Y = \alpha \quad \text{et} \quad (\varphi | Tx)_Y \leq \alpha - \varepsilon \quad \text{pour tout } x \in C.$$

Posons $l = T^* \varphi \in \mathcal{L}(T)^\perp$.

La propriété (8.4) entraîne alors :

$$C = \bigcap \{x \in X \mid (l | x)_X > \alpha - \varepsilon\} = \emptyset$$

Ce qui s'écrit encore :

$$(8.5) \quad \left(\bigcap_{i=1, \dots, m} C_i\right) \cap \{x \in X \mid (l | x)_X > \alpha - \varepsilon\} = \emptyset$$

où $C_i = \{x \in X \mid (l_i | x)_X \leq c_i\}$ $i = 1, \dots, m$.

Utilisons alors le théorème de séparation pour convexes que nous avons déjà cité dans le premier chapitre (cf Dubovickii et Miljutin [9], Lobry [21], Laurent [15] [16]) : une condition nécessaire et suffisante pour que l'égalité (8.5) soit vérifiée est qu'il existe $m+1$ nombres positifs ou nuls, non tous nuls, $\lambda, \lambda_1, \dots, \lambda_m$ vérifiant :

$$\sum_{i=1}^m \lambda_i l_i = \lambda l$$

$$\text{et} \quad \sum_{i=1}^m \lambda_i c_i \leq \lambda(\alpha - \varepsilon).$$

et l'hypothèse $C = \bigcap_{i=1, \dots, m} C_i \neq \emptyset$ entraîne $\lambda \neq 0$. Si on pose $\mu_i = \frac{\lambda_i}{\lambda}$,

on a alors :

$$(8.6) \quad \sum_{i=1}^m \mu_i l_i = l$$

$$\sum_{i=1}^m \mu_i c_i \leq \alpha - \varepsilon.$$

On peut trouver $I \subset \{1, \dots, m\}$ tel que l soit combinaison linéaire à coefficients positifs de l_i , $i \in I$ et la proposition 8.2 entraîne

$$(8.7) \quad l = \sum_{J \in \mathcal{Y}} \mu_J l_J \quad \text{avec } \mu_J \geq 0, \quad J \in \mathcal{Y}.$$

Il est aisé de vérifier que la définition de γ_J entraîne alors :

$$(8.8) \quad \sum_{J \in \mathcal{Y}} \mu_J \gamma_J = \sum_{i=1}^m \lambda_i c_i$$

La propriété $(\varphi | Tx)_Y \leq \alpha - \varepsilon$ pour tout $x \in X$ entraîne

$$(8.9) \quad \sum_{i=1}^m \lambda_i c_i \leq \alpha - \varepsilon$$

D'autre part $y \in \bigcap_{J \in \mathcal{Y}} \Omega_J$ entraîne :

$$(\varphi_J | y)_Y \leq \gamma_J \quad \text{pour tout } J \in \mathcal{Y}.$$

Soit $(\sum_{J \in \mathcal{Y}} \mu_J \varphi_J | y)_Y \leq \alpha - \varepsilon$ d'après (8.8) et (8.9) et ceci est en contradiction avec $(\varphi | y)_Y = \alpha$ puisque $\varphi = \sum_{J \in \mathcal{Y}} \mu_J \varphi_J$.

Remarque 1:

On voit aisément que le lemme 8.5 n'est pas nécessaire pour montrer la proposition 8.3. Il peut alors être considéré comme un corollaire de la proposition 8.3.

Remarque 2 :

Avec l'hypothèse (H3') au lieu de l'hypothèse (H3) nous aurions pu obtenir de la même façon les propriétés de γ_J et la caractérisation du convexe T C).

III-FORMULES DE DUALITE POUR LE PROBLEME (P).-

Définition :

\mathcal{D} désigne l'ensemble des demi-espaces fermés de X dont l'hyperplan frontière est parallèle à $\mathcal{R}(T)$: si $D = \{x \in X \mid (\ell \mid x)_X \leq \alpha\}$ avec $\ell \in X$, $\ell \neq 0$ alors ,

$$D \in \mathcal{D} \iff \ell \in \mathcal{R}(T)^\perp$$

Proposition 8.3

"On a la formule de dualité suivante :

$$\begin{aligned} d &= \min (\|Tx\|_Y \mid x \in C) \\ &= \max[\min(\|y\|_Y \mid y \in D_Y) \mid D_Y \supset \Omega] \\ &= \max[\min(\|Tx\|_Y \mid x \in D_X \mid D_X \in \mathcal{D}, D_X \supset C)] \end{aligned}$$

où D_X et D_Y désignent respectivement des demi-espaces fermés de X et de Y . "

Démonstration

On a évidemment :

$$\min(\|y\|_Y \mid y \in \Omega) \geq \max[\min(\|y\|_Y \mid y \in D_Y) \mid D_Y \supset \Omega]$$

et on obtient l'égalité pour le demi-espace :

$$\bar{D}_Y = \{y \in Y \mid (z \mid y-z)_Y \geq 0\}$$

où z est la projection de 0 sur Ω .

A chaque demi-espace $D_Y = \{y \in Y \mid (m \mid y)_Y \leq c\}$ correspond biunivoquement le demi-espace $D_X = \{x \in X \mid (\ell \mid x)_X \leq c\}$ avec $\ell = T^*m \in \text{Range}(T^*) =$

$\mathcal{R}(T)^\perp$. Si $D_Y \supset \Omega$ alors $D_X \supset C + \mathcal{R}(T) \supset C$ et inversement. L'égalité

$T(D_X) = D_Y$ entraîne alors la proposition.

C H A P I T R E IX



ALGORITHME DUAL

L'algorithme que nous décrivons dans ce chapitre est fondé sur l'étude géométrique du problème (P) que nous avons faite dans le chapitre précédent et il nous permet de résoudre le problème dual associé au problème (P). Cet algorithme provient des méthodes d'éclatement des contraintes (cf Lions et Teneau [20], Haugazeau [13], [14], Laurent et Martinet [18])

Dans le paragraphe 1 nous décrivons l'algorithme et nous en montrons la convergence au paragraphe suivant. Dans le troisième paragraphe nous étudions certains aspects pratiques de l'algorithme. Le quatrième paragraphe est consacré à des exemples numériques et à une procédure Algol.

I-DESCRIPTION DE L'ALGORITHME DUAL.-

Nous nous proposons de construire une suite de demi-espaces D^ν de Y , une suite $\{y^\nu\}$ d'éléments de Y et une suite $\{x^\nu\}$ de X telles que :

$$D^\nu \supset \Omega, \quad y^\nu \text{ est la projection de } 0 \text{ sur } D^\nu,$$

$$\lim_{\nu \rightarrow \infty} \|y^\nu\|_Y = d, \quad \lim_{\nu \rightarrow \infty} y^\nu = z = T\sigma$$

$$\text{et } Tx^\nu = y^\nu \quad \text{avec } \lim_{\nu \rightarrow \infty} x^\nu = \sigma.$$

1.1 Itération

Soit $y^\nu \in Y$. On pose :

$$D^\nu = \{y \in Y \mid (y^\nu \mid y - y^\nu)_Y \geq 0\}.$$

Supposons que y^ν et D^ν vérifient les propriétés suivantes :

$$(9.1) \quad D^\nu \supset \Omega$$

$$(9.2) \quad (\varphi_{J^\nu} \mid y^\nu)_Y = \gamma_{J^\nu} \quad \text{avec } J^\nu \in \mathcal{J}.$$

Par définition de φ_{J^ν} et de γ_{J^ν} , on peut trouver $x^\nu \in X$ tel que :

$$(9.3) \quad Tx^\nu = y^\nu \quad \text{et} \quad (l_j \mid x^\nu) = c_j \quad \forall j \in J^\nu,$$

et l'hypothèse (H4) entraîne l'unicité de x^ν .

Si $x^\nu \in C$ alors $x^\nu = \sigma$: en effet on aurait alors $y^\nu \in \Omega$ et la définition de D^ν entraîne avec la propriété (9.1) que y^ν est la projection de 0 sur Ω , donc $y^\nu = z$.

Supposons : $x^\nu \notin C$. On peut trouver $l_i^\nu \in L$ tel que, si $\varepsilon_i^\nu = (l_i \mid x^\nu)_X - c_i$ pour $i \in \{1, \dots, m\}$, on ait :

$$(9.4) \quad \varepsilon_i^\nu = \max(\varepsilon_i^\nu \mid i \in \{1, \dots, m\}) > 0.$$

Lemme 9.1 "Il existe un indice $j^\nu \in J^\nu$ tel que si l'on pose

$$(9.5) \quad J^{\nu+1} = J^\nu + \{i^\nu\} - \{j^\nu\}$$

alors $J^{\nu+1} \in \mathcal{J}$. "

Démonstration :

La démonstration de ce lemme est en fait la première partie de la démonstration du lemme 8.4.

Remarque :

La méthode d'échange proposée ici est dans son principe la même que la méthode d'échange de Stiefel utilisée dans l'algorithme de Remes (cf Laurent [17]).

Considérons $\Omega_{J^{\nu+1}}$. Par définition $y^\nu \notin \Omega_{J^{\nu+1}}$. Soit $y^{\nu+1}$ la projection de 0 sur $D^\nu \cap \Omega_{J^{\nu+1}}$. Comme la projection y^ν de 0 sur D^ν n'appartient pas à $\Omega_{J^{\nu+1}}$, $y^{\nu+1}$ est sur la frontière de $\Omega_{J^{\nu+1}}$, c'est-à-dire :

$$(9.6) \quad (\varphi_{J^{\nu+1}} \mid y^{\nu+1})_Y = \gamma_{J+1}$$

Il existe alors $x^{\nu+1}$ unique tel que :

$$Tx^{\nu+1} = y^{\nu+1} \text{ et } (l_j \mid x^{\nu+1})_X = c_j \quad \forall j \in J^{\nu+1}$$

1.2 Propriétés des suites $\{x^\nu\}$ et $\{y^\nu\}$.

$$\text{Soit } D^{\nu+1} = \{ y \in Y \mid (y^{\nu+1} \mid y - y^{\nu+1})_Y \geq 0 \}$$

Comme $D^\nu \supset D^\nu \cap \Omega_{J^{\nu+1}} \supset \Omega$, on a :

$$(9.7) \quad \|Tx^\nu\|_Y \leq \|Tx^{\nu+1}\|_Y \leq \|T\sigma\|_Y$$

ou encore $\|y^\nu\|_Y \leq \|y^{\nu+1}\|_Y \leq \|z\|_Y$.

Par définition de $y^{\nu+1}$ comme projection de 0 sur $D^\nu \cap \Omega_{J^{\nu+1}}$, on a :

$$(9.8) \quad (y^{\nu+1} \mid y - y^{\nu+1})_Y \geq 0 \quad \forall y \in D^\nu \cap \Omega_{J^{\nu+1}}$$

et comme $\Omega \subset D^\nu \cap \Omega_{J^{\nu+1}}$, on a :

$$(9.9) \quad (y^{\nu+1} \mid z - y^{\nu+1})_Y \geq 0$$

et aussi :

$$(9.10) \quad D^{\nu+1} \supset \Omega$$

Les propriétés (9.6) et (9.10) vérifiées par $y^{\nu+1}$ sont les mêmes que les propriétés (9.1) et (9.2) vérifiées par y^{ν}

1.3 Point de départ pour l'itération.

La proposition 8.3 nous donne :

$d \geq \min (\|y\|_Y \mid y \in D_Y)$ où D_Y est un demi a pour fermé contenant Ω .

Soit $J^0 \in J$. On peut prendre pour y^0 la projection de 0 sur $\mathcal{L}J^0$.

II-CONVERGENCE-

Proposition 9.1.

"Si les hypothèses (H1), (H2), (H3), ou (H'3), et (H4) sont vérifiées alors l'algorithme dual converge".

c'est-à-dire, la suite $\{x^{\nu}\}$ converge vers la solution du problème (P).

Remarque :

Si la suite $\{y^{\nu}\}$ converge vers z , la définition de $\{D^{\nu}\}$ entraîne :

$$d = \lim_{\nu \rightarrow \infty} \|y^{\nu}\| = \lim_{\nu \rightarrow \infty} \min(\|y\| \mid y \in D^{\nu}).$$

Comme $D^{\nu} \supset \Omega$, l'algorithme résoud bien alors le problème dual.

Démonstration :

(a) La suite $\{\|y^{\nu}\|_Y\}$ étant croissante et bornée par $\|z\|_Y$, elle converge vers $\alpha \leq \|z\|_Y$.

(b) Lemme 9.2 "Soit $J \in \mathcal{J}$. Si $\{u^\nu\}$ est une suite de X telle que la suite $\{T x^\nu\}$ soit bornée et que pour tout ν on ait $(l_j | x^\nu)_X = C_j$ pour $j \in J$. Alors $\{x^\nu\}$ est bornée".

Démonstration :

On pose $x^\nu = x^{1\nu} + x^{2\nu}$ avec $x^{1\nu} \in \mathcal{A}(T)$ et $x^{2\nu} \in \mathcal{A}(T)^\perp$.

Si on note \tilde{T} la restriction de T à $\mathcal{A}(T)^\perp$, \tilde{T} est une application inversible et $N(T)^\perp$ sur Y , et d'après le théorème de l'opérateur inverse \tilde{T}^{-1} est continu.

Comme $\tilde{T} u^{2\nu} = T u^{2\nu}$, $\{\|T u^{2\nu}\|_Y\}$ est bornée et, finalement, $\{u^{2\nu}\}$ est bornée.

Montrons maintenant que $\{\|u^{1\nu}\|_X\}$ est bornée.

Comme $\{u^{2\nu}\}$ est bornée, l'inégalité de Schwarz entraîne qu'il existe M tel que :

$$|(l_j | u^{2\nu})_X| < M \quad \forall j \in J, \quad \forall \nu$$

On a $(l_j | u^{1\nu}) = C_j - (l_j | u^{2\nu}) \quad \forall j \in J, \quad \forall \nu$. Il existe donc M' tel que :

$$|(l_j | u^{1\nu})| < M' \quad \forall j \in J, \quad \forall \nu$$

L'hypothèse (H4) entraîne que l'application de $\mathcal{A}^+(T)$ dans \mathbb{R}^+ qui à $\omega \in \mathcal{A}^+(T)$ associe $\max(|(l_j | \omega)_X|, j \in J) \in \mathbb{R}^+$, définit une norme sur $\mathcal{A}^+(T)$.

On en déduit que la suite $\{u^{1\nu}\}$ est bornée.

Par conséquent, la suite $\{u^\nu\}$ est bornée.

La propriété (9.3) et le lemme précédent entraînent que pour tous les indices ν vérifiant $J^\nu = J$ nous avons

$$\|x^\nu\|_X < M_J$$

Comme \mathcal{J} est fini, il existe alors M tel que

$$\|x^\nu\|_X < M \quad \forall \nu$$

Donc la suite $\{x^\nu\}$ est bornée.

On peut alors extraire de la suite $\{x^\nu\}$ une sous-suite faiblement convergente qui converge vers un élément \tilde{x} . La fonction qui à $x \in X$ associe $\|Tx\|_Y$ est convexe et continue; elle est donc faiblement continue et on en déduit

$$(9.12) \quad \|T\tilde{x}\|_Y \leq \|T\sigma\|_Y.$$

(c) Montrons maintenant que $\tilde{x} \in C$.

Comme $(y^\nu | y^{\nu+1} - y^\nu)_Y \geq 0$ (prop. (9.8)), nous avons :

$$\|y^{\nu+1} - y^\nu\|_Y^2 \leq \|y^{\nu+1}\|_Y^2 - \|y^\nu\|_Y^2.$$

Alors, d'après la partie (a) de cette démonstration,

$$(9.13) \quad \lim_{\nu \rightarrow \infty} \|y^{\nu+1} - y^\nu\|_Y = 0$$

D'autre part, comme $y^{\nu+1} \in \Omega_{J^{\nu+1}}$, nous avons

$$\|y^\nu - y^{\nu+1}\|_Y \geq \min(\|y^\nu - y\|_Y \mid y \in \Omega_{J^{\nu+1}})$$

et la définition $\Omega_{J^{\nu+1}} = \{y \in Y \mid (\varphi_{J^{\nu+1}} | y)_Y \leq \gamma_{J^{\nu+1}}\}$

entraîne :

$$\min(\|y^\nu - y\|_Y \mid y \in \Omega_{J^{\nu+1}}) = \frac{(\varphi_{J^{\nu+1}} | y^\nu)_Y - \gamma_{J^{\nu+1}}}{\|\varphi_{J^{\nu+1}}\|_Y}$$

Alors les propriétés (9.3) et (9.4) nous donnent :

$$(9.14) \quad \|y^\nu - y^{\nu+1}\|_Y \geq \frac{1}{\|\varphi_{J^{\nu+1}}\|_Y} \lambda_{i^\nu}^{J^{\nu+1}} \xi_{i^\nu}^\nu$$

Or \mathcal{J} est fini et d'après le lemme 8.3 on a :

$$\lambda_{i^\nu}^{J^{\nu+1}} > 0 \quad \forall \nu$$

Ceci entraîne qu'il existe $m > 0$ tel que :

$$\frac{\lambda_{i^\nu}^{J^{\nu+1}}}{\|\varphi_{J^{\nu+1}}\|} \geq m \quad \forall \nu$$

et, d'après les propriétés (9.13) et (9.14), on a alors :

$$\lim_{\nu \rightarrow 0} \varepsilon_{\nu}^{\nu} = 0$$

soit encore :

$$\max((l_i | \tilde{x})_X - c_i \mid i \in \{1, \dots, m\}) = 0$$

donc $\tilde{x} \in C$.

(d) Comme $\tilde{x} \in C$, on a $\|\tilde{T}\tilde{x}\|_Y \geq \|T\sigma\|_Y$. La propriété (9.12) entraîne alors $\|\tilde{T}\tilde{x}\|_Y = \|T\sigma\|_Y$ et, comme σ est l'unique solution du problème (?), on a finalement : $\tilde{x} = \sigma$.

Comme $\{\|y^{\nu}\|_Y\}$ est une suite croissante, on a alors : $\lim_{\nu \rightarrow \infty} \|y^{\nu}\|_Y = \|z\|_Y$

Or la propriété (9.9) entraîne :

$$\|z\|_Y^2 - \|y^{\nu}\|_Y^2 \geq \|z - y^{\nu}\|_Y^2 \quad \forall \nu$$

donc $\lim_{\nu \rightarrow \infty} \|y^{\nu} - z\|_Y = 0$ et y^{ν} converge fortement vers $z = T(\sigma)$

(e) Lemme (9.3) " Soit $J \in \mathcal{J}$. Si $\{x^{\nu}\}$ est une suite de X qui vérifie :

$$(l_j | x^{\nu}) = c_j \quad \forall j \in J, \quad \forall \nu$$

$$\lim_{\nu \rightarrow \infty} T(x^{\nu}) = Tx \quad \text{avec } x \in C$$

alors $\{x^{\nu}\}$ converge vers x ".

Démonstration :

Comme $\{T(x^{\nu})\}$ converge fortement vers $T(x)$, d'après un théorème dû à Banach [5], il existe une suite $\{a^{\nu}\} \in \mathcal{N}(T)$ telle que la suite $\{x^{\nu} + a^{\nu}\}$ converge fortement vers x .

On a $(l_J | x^\nu)_X = \sum_{j \in J} \lambda_j^J c_j \quad \forall \nu$ et $l_J \in \mathcal{N}(T)^\perp$.

Donc $(l_J | x)_X = \lim_{\nu \rightarrow \infty} (l_J | x^\nu + a^\nu) = \sum_{j \in J} \lambda_j^J c_j$

Comme $x \in C$, on en déduit :

$$(l_j | x)_X = c_j \quad \forall j \in J$$

et ceci entraîne :

$$(9_{-15}) \lim_{\nu \rightarrow \infty} \max_{j \in J} |(l_j | a^\nu)_X| = 0$$

Or l'application de $\mathcal{N}(T)$ dans \mathbb{R}^+ , qui à $\omega \in \mathcal{N}(T)$ fait correspondre $\max_{j \in J} |(l_j | \omega)_X|$ définit une norme sur $\mathcal{N}(T)$, espace de dimension finie. La

propriété (9₋₁₅) entraîne alors :

$$\lim_{\nu \rightarrow \infty} a^\nu = 0$$

et finalement $\lim_{\nu \rightarrow \infty} x^\nu = x$; d'où le lemme .

Soit $J \in \mathcal{J}$ tel que l'on ait $J^\nu = J$ pour un nombre infini d'indices ν . On pose

$$N_J = \{ \nu \in \mathbb{N} \mid J^\nu = J \}.$$

Alors, comme $\{ y^\nu \}$ converge fortement vers $z = T\sigma$, la propriété (9₋₃) et le lemme précédent entraînent :

$$\lim_{\substack{\nu \in N_J \\ \nu \rightarrow \infty}} x^\nu = \sigma.$$

\mathcal{J} étant fini, il existe un nombre fini de sous ensembles N_J .

On en déduit :

$$\lim_{\nu \rightarrow \infty} x^\nu = \sigma$$

III-DETERMINATION ANALYTIQUE DE L'ITERE $y^{\nu+1}$.-

3.1. Détermination directe.

$y^{\nu+1}$ est la projection de 0 sur $D^\nu \cap \Omega_{J^{\nu+1}}$ et, par définition de D^ν , y^ν est la projection de 0 sur D^ν . Soit $\tilde{y}^{\nu+1}$ la projection de 0 sur $\Omega_{J^{\nu+1}}$. Comme $y^\nu \neq y^{\nu+1}$,

deux éventualités peuvent se présenter :

(a) $\tilde{y}^{\nu+1} \in D^\nu$, c'est-à-dire :

$$(9.16) \quad (\tilde{y}^{\nu+1} - y^\nu | y^\nu)_Y \geq 0.$$

alors $\tilde{y}^{\nu+1} \in \Omega_{J^{\nu+1}} \cap D^\nu$ et $\tilde{y}^{\nu+1}$ est la projection de 0 sur $\Omega_{J^{\nu+1}} \cap D^\nu$.

On en déduit :

$$(9.17) \quad y^{\nu+1} = \tilde{y}^{\nu+1} = \lambda \varphi_{J^{\nu+1}} \quad \text{avec} \quad \lambda = \frac{\gamma_{J^{\nu+1}}}{\|\varphi_{J^{\nu+1}}\|_Y^2}.$$

(b) $\tilde{y}^{\nu+1} \notin D^\nu$.

Dans ce cas $y^{\nu+1} \in \mathcal{F}_r(\Omega_{J^{\nu+1}}) \cap \mathcal{F}_r(D^\nu)$ où $\mathcal{F}_r(D^\nu)$ (respectivement $\mathcal{F}_r(\Omega_{J^{\nu+1}})$) désigne la frontière de D^ν (resp. $\Omega_{J^{\nu+1}}$). On a alors :

$$(9.18) \quad y^{\nu+1} = \lambda \varphi_{J^{\nu+1}} + \mu y^\nu$$

où λ et μ sont solutions de système :

$$(9.19) \quad \begin{cases} \lambda \|\varphi_{J^{\nu+1}}\|_Y^2 + \mu (y^\nu | \varphi_{J^{\nu+1}})_Y = \gamma_{J^{\nu+1}} \\ \lambda (y^\nu | \varphi_{J^{\nu+1}})_Y + \mu \|y^\nu\|_Y^2 = \|y^\nu\|_Y^2. \end{cases}$$

3.2. Caractérisation de $x^{\nu+1}$.

Considérons $T^{-1}(D^\nu)$.

$$T^{-1}(D^\nu) = \{x \in X \mid (Tx^\nu - Tx \mid Tx^\nu)_X \leq 0\}$$

donc $T^{-1}(D^\nu) = \{x \in X \mid (x^\nu - x \mid T^*Tx^\nu)_X \leq 0\}$ et $T^{-1}(D^\nu)$ est un demi espace fermé de \mathcal{D} .

$$\|y^{\nu+1}\|_Y = \min(\|y\|_Y \mid y \in \Omega_{J^{\nu+1}} \cap D^\nu). \text{ Soit } y \in \Omega_{J^{\nu+1}} \cap D^\nu.$$

Il existe alors $x_1 \in C_{J^{\nu+1}}$ et $x_2 \in T^{-1}(D^\nu)$ tels que : $Tx_1 = Tx_2 = y$. Comme

$T^{-1}(D^\nu) \in \mathcal{D}$ et que $x_1 - x_2 \in \mathcal{R}(T)$ on en déduit que $x_1 \in T^{-1}(D^\nu)$. On peut alors

écrire :

$$\|Tx^{\nu+1}\|_Y = \min(\|Tx\|_Y \mid x \in C_{J^{\nu+1}} \cap T^{-1}(D^\nu)).$$

$$\text{On pose : } K_{J^{\nu+1}} = \{x \in X \mid (\ell_j \mid x)_X = c_j \quad \forall j \in J^{\nu+1}\}.$$

Par définition $x^{\nu+1} \in K_{J^{\nu+1}} \subset C_{J^{\nu+1}}$ et on peut écrire :

$$\|Tx^{\nu+1}\|_Y = \min(\|Tx\|_Y \mid x \in K_{J^{\nu+1}} \cap T^{-1}(D^\nu)).$$

Le convexe $K_{J^{\nu+1}} \cap T^{-1}(D^\nu)$ vérifiant les hypothèses (H1), (H2), (H3) et (H4'),

$x^{\nu+1}$ est la fonction "spline" unique relative à ce convexe à T.

On a pour $x^{\nu+1}$ la caractérisation suivante :

Lemme 9.4 : $x^{\nu+1}$ est la seule fonction spline relative à $K_{J^{\nu+1}} \cap T^{-1}(D^\nu)$

si et seulement si on a :

$$\frac{T^*T x^{\nu+1} - \mu T^*Tx^\nu + \lambda \ell}{J^{\nu+1}} = 0$$

avec $\lambda > 0$, $\mu > 0$ et $\mu = 0$ si $(Tx^\nu \mid Tx^{\nu+1} - Tx^\nu)_Y > 0$.

Démonstration :

D'après la proposition 1.2, on a :

$$T^* T x^{\nu+1} - T^* T x^{\nu} + \sum_{j \in J^{\nu+1}} \lambda_j \ell_j = 0 \quad \text{avec}$$

$$\lambda_j \geq 0 \quad \text{et} \quad \mu_j = 0 \quad \text{si} \quad (\ell_j | x^{\nu+1})_X < c_j$$

$$\mu \geq 0 \quad \text{et} \quad \mu = 0 \quad \text{si} \quad (T x^{\nu} | T x^{\nu+1} - T x^{\nu})_Y > 0.$$

Comme $T^*(T x^{\nu+1} - T x^{\nu}) \in \text{Range}(T^*) = \mathcal{N}(T)^{\perp}$, le lemme 8.2 et la définition de $J^{\nu+1}$ entraînent :

$$\sum_{j \in J^{\nu+1}} \lambda_j \ell_j = \lambda \ell_{J^{\nu+1}}$$

Supposons que l'on ait $\lambda = 0$. On a alors $\mu \neq 0$ c'est-à-dire

$$(T x^{\nu} | T x^{\nu+1} - T x^{\nu})_Y = 0. \quad \text{Comme} \quad \mathcal{N}(T^*) = \text{Range}(T)^{\perp} = \{0\},$$

on a aussi : $x^{\nu+1} - \mu x^{\nu} \in \mathcal{N}(T)$. Et l'égalité

$$(T x^{\nu} | T(x^{\nu+1} - \mu x^{\nu} + (1-\mu)x^{\nu}))_Y = 0$$

entraîne $\mu = 1$.

On peut alors écrire :

$$(\ell_{J^{\nu+1}} | x^{\nu+1} - x^{\nu})_X = c_{i^{\nu}} - (\ell_{i^{\nu}} | x^{\nu})_X = 0 \quad \text{ce qui est en contradiction}$$

avec la définition de i^{ν} .

Donc $\lambda \neq 0$.

L'égalité $T^* T x^{\nu+1} - \mu T^* T x^{\nu} + \lambda \ell_{J^{\nu+1}} = 0$ étant équivalente à

$T x^{\nu+1} - \mu T x^{\nu} + \lambda \varphi_{J^{\nu+1}} = 0$, on peut alors déduire du lemme 9.4 les résultats

du paragraphe 9.1.

IV-EXEMPLES NUMERIQUES ET PROCEDURES ALGOL.4.1. Exemple et aspects pratique de l'algorithme pour cet exemple.

Reprenons l'exemple du chapitre VII :

$$X = H^q = H^q [a, b] ; Y = H^0 = H^0 [a, b] , T = D^q \quad (d^p(T) = \pi^{q-1}).$$

On a encore :

$$a \leq x_1 < x_2 < \dots < x_{q+1} \leq b,$$

$$(k_i | f)_q = f(x_i) \quad i = 1, \dots, n,$$

$$L = \{l_1, \dots, l_m\} = \{k_1, \dots, k_n, -k_1, \dots, -k_n\}$$

$$C = \{f \in H^q \mid \alpha_i \leq f(x_i) \leq \beta_i, \quad i = 1, \dots, n\}.$$

On suppose que L et C vérifient les hypothèses (H1), (H2), (H3) ou (H3') et (H4).

4.1.1. Détermination de la famille \mathcal{H} .

Soit \mathcal{H} la famille des ensembles de $q + 1$ indices distincts $\in \{1, \dots, n\}$.

Soit $H = \{i_1, \dots, i_{q+1}\} \in \mathcal{H}$. (On suppose que les éléments de H sont rangés par ordre de valeur croissante). On note δ_H^q l'opérateur différence divisée d'ordre q basé sur les abscisses $x_{i_1}, \dots, x_{i_{q+1}}$. On sait que :

$$\delta_H^q = \sum_{j \in H} m_j k_j \quad \text{avec } \text{signe}(m_j) = - \text{signe}(m_{j+1}).$$

En outre, une condition nécessaire et suffisante pour que l'on ait :

$$\sum_{j \in H} \lambda_j f(x_j) = 0 \quad \text{pour tout } f \in H^q$$

est qu'il existe $\lambda \neq 0$ tel que :

$$\sum_{j \in H} \lambda_j g(x_j) = \lambda \delta_H^q (g) \quad \text{pour tout } g \in H^q.$$

A $H \in \mathcal{H}$ on peut donc faire correspondre deux éléments de \mathcal{J} que l'on notera J^+ et J^-

$$\left\{ \begin{array}{l} J^+ \subset \{1, \dots, m\} \text{ sera l'ensemble des indices associés à } \{k_1, -k_2, \dots\}; \\ J^- \text{ sera l'ensemble des indices de } \{1, \dots, m\} \text{ associés à } \{-k_1, k_2, \dots\}. \end{array} \right.$$

$$l_{J^+} = \frac{1}{\sum_{j \in H} m_j} \cdot \text{signe}(m_{i_1}) \delta_H^q \quad \text{et} \quad l_{J^-} = -l_{J^+}$$

Réciproquement, tout élément de \mathcal{Y} est formé de $q+1$ indices de $\{1, \dots, m\}$ définissant $q+1$ éléments "alternés" de L .

4.1.2. Méthode d'échange.

Soit $J^\nu \in \mathcal{Y}$ et $l_{i^\nu} \in L$ une des contraintes la moins bien vérifiée par x^ν (cf propriété (9.4)). On veut trouver $j^\nu \in J^\nu$ de sorte que $J^\nu + \{i^\nu\} - \{j^\nu\} \in \mathcal{Y}$. On va donc enlever j^ν de façon à conserver l'alternance.

Soit $l^\nu \in \{1, \dots, n\}$ l'indice associé à i^ν et H^ν le sous-ensemble de $\{1, \dots, n\}$, $H^\nu \in \mathcal{H}$, associé à J^ν .

(a) si $l^\nu > l$ pour tout $l \in H^\nu$, on considère le plus grand ou le plus petit indice de H^ν suivant le "type" de la contrainte définie par i^ν . On enlève l indice $j^\nu \in J^\nu$ correspondant. On fait de même si $l^\nu < l$ pour tout $l \in H^\nu$.

(b) sinon, on considère les deux indices qui "encadrent" l^ν . On enlève celui des deux indices correspondants de J^ν qui définit une contrainte de "même type" que la contrainte définie par i^ν .

4.1.3. Itération.

Soit $H^0 \in \mathcal{H}$, J^0 l'un ou l'autre des éléments de \mathcal{Y} qui lui correspondent et y^ν la projection de 0 sur Ω_{J^0} . Alors, d'après le paragraphe 3.2 de ce chapitre,

x^0 est la fonction "spline" relative à K_J^0 . On sait alors que x^0 est formé de morceaux de polynômes de degré $2q-1$ qui se raccordent aux points x_i , $i \in H^0$ jusqu'à la dérivée $2q-2$ ^{ième} comprise. On définit J^1 et H^1 par la méthode d'échange. D'après les propriétés (9.17) ou (9.18), on a alors :

$$Tx^1 = \mu_1 Tx^0 + \lambda_1 \varphi_{J^1} \quad \text{avec } T^* \varphi_{J^1} = \ell_{J^1}.$$

Etant donnée la propriété de ℓ_{J^1} établie dans le paragraphe précédent, on peut montrer (cf Laurent et Anselone [1]) que φ_{J^1} est formée de morceaux de polynômes de degré $q-1$ qui se raccordent aux points d'abscisses x_i , $i \in H^1 \cup H^0$ jusqu'à la dérivée $q-2$ ^{ième} comprise. On en déduit que, quel que soit ν , x^ν est formée de morceaux de polynômes de degré $2q-1$ se raccordant jusqu'à la dérivée $2q-2$ ^{ième} comprise aux points d'abscisse x_i , $i = 1, \dots, n$.

Formules.

Soit $z^\nu \in \mathbb{R}^n$, le vecteur des sauts de dérivée $2q-1$ ^{ième} de la fonction x^ν aux points d'abscisse x_i , $i = 1, \dots, n$. On a alors :

$$T^* Tx^\nu = \sum_{i=1}^n z_i^\nu k_i$$

D'autre part, $\|y^\nu\|_0^2 = (Tx^\nu | Tx^\nu)_0 = (x^\nu | T^* Tx^\nu)_q$

$$\text{Donc} \quad \|y^\nu\|_0^2 = \sum_{i=1}^n z_i^\nu x^\nu(x_i).$$

On a aussi (cf Carasso [6]) :

$$(\varphi_{J^\nu} | \varphi_{J^\nu})_q = \frac{(-1)^q}{(2q-1)!} \delta_{J^\nu}(h(t))$$

$$\text{avec } h(t) = \delta_{J^\nu} (t-x)_+^{2q-1}.$$

4.2. Procédures Algol.Notice d'emploi des procédures CØØRQQ et ØRDEC2 :

COORQQ (X, Y, SAU, IN, N, Q)

ORDERC (X, Y, SAU, IN, N, Q)

Etant donnés les entiers N et Q, ces procédures calculent les ordonnées $Y[I]$, $J = 1, \dots, N$ de la fonction formées de morceaux de polynômes de degré $2 \times Q - 1$ se raccordant aux points d'abscisses $X[I]$, $I = 1, \dots, N$, jusqu'à la dérivée $2Q - 2$ ^{ième} comprise, ayant en ces points des sauts de dérivée $2Q - 1$ ^{ème} donnés par le tableau $SAU[1 : N]$ et passant par $Q + 1$ points définis par le tableau entier $IN[1 : Q + 1]$ (IN contient les indices définissant les abscisses de ces points).

La procédure CØØRQQ utilise la formule (cf Carasso [6]) :

$$Y[I] = \mathcal{P}_{q-1}(X[I]) + \sum_{J=1}^{I-1} SAU[J] (X[I] - X[J])^{2q-1}$$

où \mathcal{P}_{q-1} désigne un polynôme de degré $q - 1$.

La procédure ØRDEC 2 s'utilise dans le cas où $Q = 2$. Elle est basée sur la même idée que les "méthodes d'intégration locales" de Carasso.

Notice d'emploi de la procédure CONV DUAL.

CONVDUAL(Q, N, X, Y, SAU, A, B, X1, JN, EPS, ETIQ);

Etant donnés les entiers Q et N, cette procédure calcule au moyen de l'algorithme dual les ordonnées $Y[I]$, $I = 1, \dots, N$. et les sauts de dérivée $2Q - 1$ ^{ième}, $SAU[I]$, $I = 1, \dots, N$ de la fonction "spline" relative au convexe:

$$C = \{f \in H^Q \mid A[I] \leq f(X[I]) \leq B[I], I = 1, \dots, N\}.$$

et à l'application $T = D^Q$.

Le tableau $X1[1 : Q + 1]$ et le tableau entier $JN[1 : Q + 1]$ doivent contenir, à l'appel de la procédure, les abscisses et les indices correspondants qui définissent x^ν .

On voit par l'étiquette ETIQ si le point de départ a été mal choisi : c'est-à-dire si $\|Tx^0\|_0 = 0$.

On arrête les calculs dès que : $\text{Max}(Y[I] - B[I], A[I] - Y[I]) < \text{EPS}$.

4.3. Exemple numérique.

Reprenons l'exemple numérique du chapitre VII :

$$q = 2, n = 10, f(x) = \sin x, \quad x_i = i-1 \quad (i = 1, \dots, n).$$

Soit $\epsilon = 0.5$. Désignons par $+i$ les points x_i où $\sigma(x_i) = f(x_i) + \epsilon$ et par $-i$ les points où $\sigma(x_i) = f(x_i) - \epsilon$. On peut alors exprimer la solution σ par :

$$\{ +1, -3, +6, -8, +10 \}.$$

On a aussi $\|T\sigma\|_0^2 = 0.33702\dots$

Résultats dans le cas où $J^0 = \{-3, +6, -9\}$

ν	J^ν			$\ Tu^\nu\ _0^2$	i_ν	
0	-3	6	-9	0.18331	0.79086	
1	1	-3	6	0.28474	0.14678	
2	-3	6	-8	0.29241	0.34202	
3		6	-8	10	0.31816	0.28155
4	1		-8	10	0.31886	0.09550
5	1	-3		10	0.32016	0.05725
6	1	-3	6		0.32040	0.05269
7		-3	6	-8	0.32147	0.07591

Temps d'exécution = 7,2 s.

On a une précision relative de 3.10^{-2} après 25 itérations.

Si au lieu de prendre $\epsilon = 0.5$ on avait pris $\epsilon = 0.1$, avec le même point de départ, on aurait une précision relative de 8.10^{-2} seulement, après 25 itérations: plus les "bâtonnets" sont larges, plus la convergence est rapide.

La rapidité de la convergence dépend aussi évidemment du point de départ choisi : si on prend $J^0 = \{1, -4, 7\}$ il faut 113 itérations au moins dans le cas $\epsilon = 0.1$ pour avoir la précision relative de 3.10^{-2} .


```

PROCEDURE CONVQUAL(Q,N,X,Y,SAU,A,B,X1,JN,EPS,ETIQ);
  USE Q,N,X,A,B,EPS;
  INTEGER Q,N;
  REAL X,Y,SAU,A,B,X1;
  AL ARRAY JN;
  AL EPS;
  REL ETIQ;

```

```

BEGIN
  REAL PROCEDURE DIFDIV(X,Y,K);
  VALUE X,Y,K;
  ARRAY X,Y;
  INTEGER K;

```

```

  BEGIN
    INTEGER I,J;
    FOR J:=1 STEP 1 UNTIL K DO FOR I:=1 STEP 1
      UNTIL K-J+1 DO Y(/I/):= IE ABS(Y(/I+1/)-Y(/I/))=0
      THEN 0 ELSE (Y(/I+1/)-Y(/I/))/(X(/I+J/)-X(/I/));
    DIFDIV:=Y(/I/)
  END DIFDIV;

```

```

  REAL PROCEDURE H(X,T,K);
  ARRAY X;
  REAL T;
  INTEGER K;

```

```

  BEGIN
    ARRAY L(/1:K+1/);
    INTEGER I,J;
    REAL U,S;
    IF T>X(/1/) THEN
      BEGIN
        FOR I:=1 STEP 1 UNTIL K+1 DO IE X(/I/)<T
          THEN
            BEGIN
              U:=S:=(T-X(/I/))*X(/I/);
              FOR J:=1 STEP 1 UNTIL K-2 DO S:=S*U;
              S:=S*(T-X(/I/));
              L(/I/):=S
            END
          ELSE L(/I/):=0;
        FOR J:=1 STEP 1 UNTIL K DO FOR I:=1 STEP 1
          UNTIL K-J+1 DO L(/I/):= IE ABS(L(/I+1/)-L(/I/))=0
          THEN 0 ELSE (L(/I+1/)-L(/I/))/(X(/I+J/)-X(/I/));
        S:=L(/1/);
        FOR I:=2 STEP 1 UNTIL 2*K-1 DO S:=S/I;
        H:=S
      END
    ELSE H:=0
  END H;

```



```

PROCEDURE ORDEC2(X,Y,SAU,IN,N,Q);
ARRAY X,Y,SAU;
INTEGER N,Q;
INTEGER ARRAY IN;

BEGIN
  INTEGER I,I1,I2;
  REAL Z,H,A2,A3,A4;
  ARRAY DY,D2Y,D3YM,D3YP(/1:N/);
  I1:=IN(/1/);
  I2:=IN(/2/);
  DERIV:=D2Y(/1/):=D3YM(/1/):=D3YP(/N/):=D2Y(/N/):=0;
  FOR I:=1 STEP 1 UNTIL ENTIER(N/2)-1 DO
    BEGIN
      D3YP(/I/):=D3YM(/I+1/):=D3YM(/I/)+SAU(/I/);
      D2Y(/I+1/):=D2Y(/I/)+(X(/I+1/)-X(/I/))*D3YP(/I/);
    END;
  FOR I:=N STEP -1 UNTIL ENTIER(N/2)+1 DO
    BEGIN
      D3YM(/I/):=D3YP(/I-1/):=D3YP(/I/)-SAU(/I/);
      D2Y(/I-1/):=D2Y(/I/)+(X(/I-1/)-X(/I/))*D3YM(/I/);
    END;
  I:=I1;
  A2:=A3:=A4:=0;
  Z:=Y(/I1/);
  ALLER:H:=X(/I+1/)-X(/I/);
  A4:=-H*H*H/6+H*H*A2/2-H*A3+A4;
  A3:=-H*H/2-H*A2+A3;
  A2:=-H+A2;
  IF I>=I2-1 THEN GOID RETOUR;
  Z:=A4*SAU(/I+1/)+Z;
  I:=I+1;
  GOID ALLER;
  RETOUR:DY(/I2/):=(Z-Y(/I2/)-A3*D2Y(/I2/)-A4*D3YM(/I2/))/A2;
  ;
  IF I2>1 THEN FOR I:=I2-1 STEP -1 UNTIL 1 DO

    BEGIN
      H:=X(/I/)-X(/I+1/);
      Y(/I/):=Y(/I+1/)+H*(DY(/I+1/)+H*(D2Y(/I+1/)+H*D3YM(/I+1/)/3)/2);
      DY(/I/):=DY(/I+1/)+H*(D2Y(/I+1/)+H*D3YM(/I+1/)/2);
    END;
  IF I2<N THEN FOR I:=I2+1 STEP 1 UNTIL N DO

    BEGIN
      H:=X(/I/)-X(/I-1/);
      Y(/I/):=Y(/I-1/)+H*(DY(/I-1/)+H*(D2Y(/I-1/)+H*D3YP(/I-1/)/3)/2);
      DY(/I/):=DY(/I-1/)+H*(D2Y(/I-1/)+H*D3YP(/I-1/)/2)
    END;
  END ORDEC2;

```



```

PROCEDURE SPLIN2(T,Y,N,DER);
ARRAY T,Y,DER;
INTEGER N;
REAL PROCEDURE POLINT(K,N,X,D,T);
COMMENT VOIR L'ANNEXE DE PROCEDURES;
INTEGER I,J,L1,K1,K0,K,L;
REAL E,E1,TJ2,DU,DUI,PHI2,R,LAMBDA,S,D1,D2,MU;
REAL ARRAY Y1,Y2(/I:Q+1/), EOB I:=1 STEP 1 UNTIL N
DO IN(/I/):=0;
J:=1;
FOR I:=1 STEP 1 UNTIL Q+1 DO
  BEGIN
    Y1(/I/):= IF J=1 THEN A(/JN(/I//)) ELSE B(/JN(/I//));
    Y2(/I/):= IF J=1 THEN B(/JN(/I//)) ELSE A(/JN(/I//));
    J:=J*(-1);
  END ;
  D1:=DIFDIV(X1,Y1,Q);
  D2:=DIFDIV(X1,Y2,Q);
  IF SIGN(D1)*SIGN(D2)<0 THEN GOIQ ETIQ;
  IF ABS(D1)>ABS(D2) THEN
    BEGIN
      J:=+1;
      FOR I:=1 STEP 1 UNTIL Q+1 DO Y1(/I/):=Y2(/I/)
    END ;
  IF ABS(D1)<ABS(D2) THEN J:=-1;
  FOR I:=1 STEP 1 UNTIL Q+1 DO
    BEGIN
      JN(/I/):=JN(/I/)*J;
      J:=J*(-1);
      IN(/ABS(JN(/I//))):=SIGN(JN(/I/))
    END ;
  GO IQ CALI;
  ITER:I:=1;
  J:=0;
  L1:=0;
  E:=EPS;
  ECHANGE: IF IN(/I/)-=0 THEN
    BEGIN
      K1:=I;
      L1:=L1+1;
      IF L1=1 THEN K0:=K1;
      SUIQ RECOM
    END ;
  IF IN(/I/)=0 THEN
    BEGIN
      IF A(/I/)>Y(/I/) THEN
        BEGIN

```



```

      E1:=A(/I/)-Y(/I/);
      IF E1>E THEN
        BEGIN
          J:=-I;
          E:=E1;
          L:=L1;
          IF L=0 THEN K:=K1;
        END ;
      END ;
      IF Y(/I/)>B(/I/) THEN
        BEGIN
          E1:=Y(/I/)-B(/I/);
          IF E1>E THEN
            BEGIN
              J:=I;
              E:=E1;
              L:=L1;
              IF L=0 THEN K:=K1;
            END ;
          END ;
        END ;
      RECOM: IF I=N THEN GO TO CHOIX;
      I:=I+1;
      GO TO FCHANGE;
      CHOIX: IF J=0 OR E<EPS THEN GO TO TERM;
      IF (L=0)&(L=Q+1) THEN
        BEGIN
          IF SIGN(J)=SIGN(IN(/K/)) THEN IN(/K/):=0 ELSE
            FOR I:=K+1 STEP 1 UNTIL N DO
              BEGIN
                IF IN(/I/)=0 THEN
                  BEGIN
                    IN(/I/):=0;
                    GO TO JO
                  END ;
                END ;
              END ;
          IF L=0 THEN
            BEGIN
              IF SIGN(J)=SIGN(IN(/K0/)) THEN IN(/K0/):=0 ELSE IN(/K1/):=0
            END ;
          IF L=Q+1 THEN
            BEGIN
              IF SIGN(J)=SIGN(IN(/K1/)) THEN IN(/K1/):=0 ELSE IN(/K0/):=0
            END ;
          JO:=IN(/ABS(J)/):=SIGN(J);
          TU2:=0;
          FOR I:=1 STEP 1 UNTIL N DO TU2:=TU2+SAU(/I/)*Y(/I/);
          J:=1;
          FOR I:=1 STEP 1 UNTIL N DO

```



```

BEGIN
  IF IN(/I/)-=0 THEN
    BEGIN
      UN(/J/):=I;
      XI(/J/):=X(/I/);
      YI(/J/):=Y(/I/);
      J:=J+1;
    END ;
  END ;
  CU:=DIFDIV(X1,Y1,Q);
  J:=1;
  FOR I:=1 STEP 1 UNTIL N DO
    BEGIN
      IF IN(/I/)-=0 THEN
        BEGIN
          YI(/J/):= IF IN(/I/)=1 THEN B(/I/) ELSE A(/I/);
          J:=J+1;
        END ;
      END ;
      DU1:=DIFDIV(X1,Y1,Q);
      FOR I:=1 STEP 1 UNTIL Q+1 DO Y2(/I/):=H(X1,XI(/I/),Q);
      PHI2:=ABS(DIFDIV(X1,Y2,Q));
      TEST:R:=DU1*DU-TU2*PHI2;
      IF R<0 THEN GO TO CAL2;
      CAL1:SPL IN(X1,Y1,Q+1,DER);
      J:=0;
      FOR I:=1 STEP 1 UNTIL N DO
        BEGIN
          IF IN(/I/)=0 THEN
            BEGIN
              SAU(/I/):=0;
              Y(/I/):=POLINT(Q,Q+1,X1,DER,X(/I/));
            END
          ELSE
            BEGIN
              J:=J+1;
              SAU(/I/):=DER(/J,2*Q-1/);
              Y(/I/):=YI(/J/);
            END ;
          END ;
          GO TO ITER;
        CAL2:LAMBDA:=TU2*PHI2-DU*DU;
        MU:=-R/LAMBDA;
        LAMBDA:=(DU1-DU)*TU2/LAMBDA;
        FOR I:=1 STEP 1 UNTIL N DO SAU(/I/):=MU*SAU(/I/);
        J:=0;
        FOR I:=1 STEP 1 UNTIL N DO
          BEGIN
            IF IN(/I/)-=0 THEN
              BEGIN

```



```
J:=J+1;  
Y(/I/):=YI(/J/);  
S:=1;  
FOR K:=1 STEP 1 UNTIL J-1, J+1 STEP 1 UNTIL Q+1  
DO S:=S/(XII(/J/)-XII(/K/));  
SAU(/I/):=SAU(/I/)+LAMBDA*S;  
END ;  
END ;  
ORDEC2(X,Y,SAU,JN,N,C);  
GO TO ITER;  
TERM;  
END  
ENC ;
```



```

PROCEDURE COORQ(X,Y,SAU,IN,N,Q);
ARRAY X,Y,SAU;
INTEGER ARRAY IN;
INTEGER N,Q;

BEGIN
  INTEGER I,J,K;
  REAL S;
  ARRAY A(/1:Q/),B(/1:Q/),G(/1:Q,1:Q/);
  IF IN(/1/)=1 THEN B(/1/):=Y(/1/);
  FOR J:=( IF IN(/1/)=1 THEN 2 ELSE 1) STEP 1 UNTIL Q
  DO
    BEGIN
      B(/J/):=0;
      FOR I:=1 STEP 1 UNTIL IN(/J/)-1 DO
        BEGIN
          S:=SAU(/I/);
          FOR K:=1 STEP 1 UNTIL 2*Q-1 DO S:=S*(X(/IN(/J//))-X(/I/
          ))/K;
          B(/J/):=B(/J/)+S
        END ;
      B(/J/):=Y(/IN(/J//))-B(/J/)
    END ;
    FOR I:=1 STEP 1 UNTIL Q DO
      BEGIN
        G(/I,1/):=1;
        FOR J:=1 STEP 1 UNTIL Q-1 DO G(/I,J+1/):=G(/I,J/)*X(/IN(/I
        //))
      END ;
      GRESOLSYSLINE(G,B,A,Q,TERM);
      FOR I:=1 STEP 1 UNTIL N DO
        BEGIN
          Y(/I/):=A(/Q/);
          FOR J:=1 STEP 1 UNTIL Q-1 DO Y(/I/):=Y(/I/)*X(/I/) + A(/Q-J/
          )
        END ;
      FOR I:=2 STEP 1 UNTIL N DO FOR J:=1 STEP 1 UNTIL I-1
      DO
        BEGIN
          S:=SAU(/J/);
          FOR K:=1 STEP 1 UNTIL 2*Q-1 DO S:=S*(X(/I/)-X(/J//))/K;
          Y(/I/):=Y(/I/)+S
        END ;
      TERM:
    END COORQ;

```


ANNEXE : Procédures Algol



1) Procédure GRESOLSYSLINE (A, B, X, N, IMPOSSIBLE).

Cette procédure résoud le système linéaire $AX = B$ (où A est une matrice carré d'ordre N) par la méthode de Gauss. Si la matrice A est singulière, on sort par l'étiquette IMPOSSIBLE.

2) Procédure POLINT (K, N, X, D, T) ; voir le commentaire inséré dans la procédure (cf Carraso [6]).

3) Procédure INSPLIN RELAQ ou SPLIN (T, Y, Q, N, DER) ; voir le commentaire inséré dans la procédure.

Cette procédure utilise la méthode du "transport de relations" (cf Carasso et Laurent [7]).

Dans le cas $Q = 2$, la méthode s'écrit plus simplement (voir procédure SPLDEU).

4) Procédure SPLUN (Y, N, MU) ;

Cette procédure calcule le vecteur MU des sauts des dérivées de la fonction spline d'ordre 1 passant par les points $(X[I], Y[I])$ (On déclare le tableau $X[1 : N]$ dans un bloc contenant SPLUN).

5) Procédure SPLDEU (Y, N, MU) ;

Elle calcule des sauts de dérivée des fonctions spline d'interpolation d'ordre 2 au moyen de la méthode du "transport des relations".

6) Procédures SAUDERIV (K, N, X, Y, G, MU), CØNSTR(G, K, N, X), FI(X, T, X, K),
DIF DIV(J, L, X, L);

Ces procédures permettent de calculer les sauts de dérivée d'une fonction "spline" d'interpolation d'ordre K passant par les points $(X[I], Y[I])$, $I = 1, \dots, N$ (cf Carasso [6]).

6) procédure SAUBASE (X, SAU, N);

Etant donné les N points $X[I]$, $I = 1, \dots, N$, cette procédure calcule dans le tableau $SAU[1 : N, 1 : N]$ les sauts de dérivée des N fonctions "spline" de relatives à ces points. (Nous avons ici le cas des "spline" d'interpolation d'ordre 2).

1. PROCEDURE GRESOLSYSLINE(A,B,X,N,IMPOSSIBLE);
 ARRAY A,B,X;
 INIEGER N;
 LABEL IMPOSSIBLE;

ANNEXE

BEGIN
 TRIANGULARISATION:

BEGIN

INIEGER I,J,K;

REAL R;

FOR K:=1 STEP 1 UNTIL N-1 DO

BEGIN

NORMAL:

BEGIN

IF ABS(A(/K,K/))=0 THEN GOTO ECHANGEDELIGNES;

FOR I:=K+1 STEP 1 UNTIL N DO

BEGIN

R:=A(/I,K/)/A(/K,K/);

FOR J:=K+1 STEP 1 UNTIL N DO A(/I,J/):=A(/I,J/)-R
 *A(/K,J/);

B(/I/):=B(/I/)-R*B(/K/)

END

END ;

GOTO RETOUR;

ECHANGE DE LIGNES:

BEGIN

INIEGER M;

M:=K+1;

RET:

IF ABS(A(/M,K/))=0 THEN M:=M+1 ELSE GOTO CONT;

IF M=N+1 THEN GOTO IMPOSSIBLE ELSE GOTO RET;

CONT:

FOR J:=K STEP 1 UNTIL N DO

BEGIN

R:=A(/K,J/);

A(/K,J/):=A(/M,J/);

A(/M,J/):=R

END ;

R:=B(/K/);

B(/K/):=B(/M/);

B(/M/):=R;

GOTO NORMAL

END ;

RETOUR:

END

END TRIANGULARISATION;

RESSYSTRI:

BEGIN

INIEGER I,J;

REAL TX;

FOR I:=N STEP -1 UNTIL 1 DO

BEGIN

TX:=0;

FOR J:=N STEP -1 UNTIL I+1 DO TX:=TX-X(/J/)*A(/I,J/);

IF A(/I,I/)=0 THEN GOTO IMPOSSIBLE;

X(/I/):=(B(/I/)+TX)/A(/I,I/)

END

END RESSYSTRI

END GRESOLSYSLINE;

2.

```

REAL PROCEDURE POL INT(K,N,X,D,T);
VALUE K,N,X,D,T;
INTEGER K,N;
ARRAY X,D;
REAL T;
COMMENT

```

CETTE PROCEDURE CALCULE LA VALEUR QUE PREND A L'ABSCISSE T LA SPLINE D'INTERPOLATION(K) ADMETTANT AUX POINTS X(I/I) (1 <= I <= N) LES DERIVEES D(I/I,J/I) (0 <= J <= K-1) (K=2,3,4)

```

BEGIN
REAL PROCEDURE Q(P,K,A,B,X);
VALUE A,B,P,K,X;
REAL A,B,X;
INTEGER P,K;
IF K=2 THEN
  BEGIN
    IF P=0 THEN Q:=(1-X)*(1-X)*(1+2*X) ELSE Q:=(1-X)*(1-X)*X*(
      B-A)
    END
  ELSE IF K=3 THEN
    BEGIN
      IF P=0 THEN Q:=(1-X)*(1-X)*(1-X)*((6*X+3)*X+1)
      ELSE IF P=1 THEN Q:=X*(1-X)*(1-X)*(1-X)*(1+3*X)*(B-A)
      ELSE Q:=5*(1-X)*(1-X)*(1-X)*X*X*(B-A)*(B-A)
      END
    ELSE IF K=4 THEN
      BEGIN
        IF P=0 THEN Q:=(1-X)*(1-X)*(1-X)*(1-X)*(((20*X+10)*X+4)*X+
          1) ELSE IF P=1 THEN Q:=X*(1-X)*(1-X)*(1-X)*(1-X)*(B-A)*((
          10*X+4)*X+1) ELSE IF P=2 THEN Q:=5*X*X*(1-X)*(1-X)*(1-X)
          *(1-X)*(B-A)*(B-A)*(1+4*X) ELSE Q:=X*X*X*(1-X)*(1-X)*(1-X)
          *(1-X)*(B-A)*(B-A)*(B-A)/6
        END Q;
        INTEGER P,I;
        REAL A,H,S,D;
        IF T<=X(I/I) THEN GO TO SUITE;
        I:=1;
        JO: IF T<=X(I+1/I) THEN GO TO SUITE ELSE
          BEGIN
            I:=I+1;
            IF I>=N THEN GO TO SUITE;
            GO TO JO;
          END ;
        SUITE:H:=T-X(I/I);
        S:=D(I,I,K-1/I);
        FOR I:=K-1 STEP -1 UNTIL 1 DO S:=S*H/I+D(I,I-1/I);
        GO TO FIN;
        SUITE:H:=T-X(I/N/I);
        S:=D(I,N,K-1/I);
        FOR I:=K-1 STEP -1 UNTIL 1 DO S:=S*H/I+D(I,N,I-1/I);
        GO TO FIN;
        SUITE:H:=X(I+1/I)-X(I/I);
        A:=T-X(I/I);
        B:=X(I+1/I)-T;
        S:=0;
        FOR P:=0 STEP 1 UNTIL K-1 DO S:=S+Q(P,K,X(I/I),X(I+1/I),A/H)*
          D(I,I,P/I)+( IF P/I+2=P THEN 1 ELSE -1)*Q(P,K,X(I/I),X(I+1/I),
          B/H)+D(I+1,P/I);
        FIN: POL INT:=S
      END
    END
  END
END

```



```

3. PROCEDURE INSPI (M, E, A, O, T, Y, Q, N, DER);
  ARRAY T, Y, DER;
  INTEGER Q, N;
  COMMENT  CETTE PROCEDURE CALCULE LES DERIVÉES
  DER(I, J) (1 ≤ I ≤ N) (1 ≤ J ≤ 2*Q-2)
  DE LA SPLINE D'INTERPOLATION DÉFINIE PAR
  X ET Y(I: N);

  BEGIN
    ARRAY M(I: N, 1: Q-1, 1: 2*Q-2), S(I: N, 1: Q-1), M1(I: Q-1, 1: 2*Q)
    , S1(I: Q-1), A(I: 2*Q-2, 1: 2*Q-2), B, X(I: 2*Q-2), ST(I: 2*Q);
    REAL H, P, PK;
    INTEGER I, J, K, K1;
    PHASE1: I:=1;
    FOR J:=1 STEP 1 UNTIL Q-1 DO
      BEGIN
        FOR K:=1 STEP 1 UNTIL 2*Q DO M(I, J, K):= 0;
        IF I=1 THEN M(I, J, K):= Y(I);
        S1(I, J):=0;
      END ;
    ITER1: FOR J:=1 STEP 1 UNTIL Q-1 DO
      BEGIN
        FOR K:=1 STEP 1 UNTIL 2*Q-2 DO M(I, J, K):=M(I, J, K+1);
        S(I, J):=S1(I, J)-M(I, J, 1)*Y(I);
      END ;
    IF I=N THEN GOTO PHASE2;
    H:=T(I+1)-T(I);
    FOR J:=1 STEP 1 UNTIL Q-1 DO
      BEGIN
        FOR K:=1 STEP 1 UNTIL 2*Q DO
          BEGIN
            P:=0;
            FOR K1:=1 STEP 1 UNTIL K DO P:=M(I, J, K1)-P*H/(K-K1+1);
            ST(K):=P;
          END ;
        FOR K:=1 STEP 1 UNTIL 2*Q DO M(I, J, K):=ST(K);
      END ;
    FOR J:=Q-1 STEP -1 UNTIL 2 DO
      BEGIN
        FOR K:=1 STEP 1 UNTIL 2*Q-1 DO M(I, J, K):=M(I, J, K)*M(I, J-1, 2*Q)-M(I, J-1, K)-M(I, J, 2*Q);
        S1(I, J):=S1(I, J)-M(I, J-1, 2*Q)-S1(I, J-1)-M(I, J, 2*Q);
        M(I, J, 2*Q):=0;
      END ;
    P:=1;
    FOR K:=1 STEP 1 UNTIL 2*Q-1 DO P:=P*(-H)/K;
    FOR K:=1 STEP 1 UNTIL 2*Q-1 DO
      BEGIN
        PK:=1;
        FOR K1:=2 STEP 1 UNTIL K DO PK:=PK*(-H)/(K1-1);
        M1(I, K):=M(I, K)-P-PK*M(I, 2*Q);
      END ;
    S1(I):=S1(I)+P-Y(I)-M1(I, 2*Q);
    M1(I, 2*Q):=0;
    FOR J:=1 STEP 1 UNTIL Q-1 DO
      BEGIN
        PK:=S1(I, J)-S1(I, J+1);
        FOR K:=1 STEP 1 UNTIL 2*Q-1 DO PK:=PK+M1(I, J, K)-M1(I, J+1, K);
        PK:=SORT(PK);
      END ;
    END ;
  END ;

```



```

S1(/J/):=S1(/J/)/PK;
FOR K:=1 STEP 1 UNTIL 2*Q-1 DO M1(/J,K/):=M1(/J,K/)/PK
END ;
I:=I+1;
GOTO ITER1;
PHASE2: I:=N;
FOR J:=1 STEP 1 UNTIL Q-1 DO
  BEGIN
    FOR K:=1 STEP 1 UNTIL 2*Q DO M1(/J,K/):= IF K=Q+J
      THEN 1:0 ELSE 0;
    S1(/J/):=0
  END ;
ITER2: FOR J:=1 STEP 1 UNTIL Q-1 DO
  BEGIN
    FOR K:=1 STEP 1 UNTIL 2*Q-2 DO
      BEGIN
        A(/J,K/):=M(/I,J,K/);
        A(/Q-1+J,K/):=M1(/J,K+1/)
      END ;
      B(/J/):=S(/I,J/);
      B(/Q-1+J/):=S1(/J/)-Y(/I/)*M1(/J,1/)
    END ;
    GRESOL (A,B,X,2*Q-2,TERM);
    FOR J:=1 STEP 1 UNTIL 2*Q-2 DO DER(/I,J/):=X(/J/);
    IF I=1 THEN GOTO TERM;
    H:=T(/I/)-T(/I-1/);
    FOR J:=1 STEP 1 UNTIL Q-1 DO
      BEGIN
        FOR K:=1 STEP 1 UNTIL 2*Q DO
          BEGIN
            P:=0;
            FOR K1:=1 STEP 1 UNTIL K DO P:=M1(/J,K1/)+P*H/(K-K1+1);
            ST(/K/):=P
          END ;
          FOR K:=1 STEP 1 UNTIL 2*Q DO M1(/J,K/):=ST(/K/)
        END ;
        FOR J:=Q-1 STEP 1 UNTIL 2 DO
          BEGIN
            FOR K:=1 STEP 1 UNTIL 2*Q-1 DO M1(/J,K/):=M1(/J,K/)*M1(/J-
              1,2*Q/)-M1(/J-1,K/)*M1(/J,2*Q/);
            S1(/J/):=S1(/J/)*M1(/J-1,2*Q/)-S1(/J-1/)*M1(/J,2*Q/);
          END ;
          P:=1;
          FOR K:=1 STEP 1 UNTIL 2*Q-1 DO P:=P+H/K;
          FOR K:=1 STEP 1 UNTIL 2*Q-1 DO
            BEGIN
              PK:=PK*H/(K1-1);
              M1(/1,K/):=M1(/1,K/)*P-PK*M1(/1,2*Q/)
            END ;
            S1(/1/):=S1(/1/)*P-Y(/1/)*M1(/1,2*Q/);
            M1(/1,2*Q/):=0;
          FOR J:=1 STEP 1 UNTIL Q-1 DO
            BEGIN
              PK:=S1(/J/)*S1(/J/);
              FOR K:=1 STEP 1 UNTIL 2*Q-1 DO PK:=PK+M1(/J,K/)*M1(/J,K/);
              PK:=SQRT(PK);
              S1(/J/):=S1(/J/)/PK;
              FOR K:=1 STEP 1 UNTIL 2*Q-1 DO M1(/J,K/):=M1(/J,K/)/PK;
            END ;
          I:=I-1;
          GOTO ITER2;
        TERM:

```



```

4. PROCEDURE SPLUN(X,Y,N,MU);
   ARRAY X,Y,MU;
   INTEGER N;

   BEGIN
     INTEGER I;
     FOR I:=1 STEP 1 UNTIL N-1 DO MU(I/):=- (Y(I+1/)-Y(I/))/(X(I+1/)-X(I/));
     MU(N/):=-MU(N-1/);
     IF N=2 THEN GOIQ TERM;
     FOR I:=N-1 STEP -1 UNTIL 2 DO MU(I/):=MU(I/)-MU(I-1/);
     TERM;
   END SPLUN;

```

```

5. PROCEDURE SPLCEU(N,Y,MU);
   INTEGER N;
   ARRAY Y,MU;

   BEGIN
     REAL M1,M2,M3,S,H,F1,F2,F3,F4,P,A2,B2,C2,DEN;
     INTEGER I;
     ARRAY A1,B1,C1,DY,D2Y,D3YP,D3YM(I:N/);
     PHASE1:M1:=0;
     M2:=0;
     M3:=1;
     S:=0;
     I:=1;
     ITER1:A1(I/):=M2;
     B1(I/):=M3;
     C1(I/):=S-M1*Y(I/);
     IF I=N THEN GOIQ PHASE2;
     H:=T(I+1/)-T(I/);
     F1:=M1;
     F2:=M1*(-H)+M2;
     F3:=M1*H*H/2+M2*(-H)+M3;
     F4:=M1*(-H*H*H/6)+M2*H*H/2+M3*(-H);
     P:=-H*H*H/6;
     M1:=F1*P-F4;
     M2:=F2*P-F4*(-H);
     M3:=F3*P-F4*H*H/2;
     S:=S*P-F4*Y(I/);
     DEN:=SQRT(M1*M1+M2*M2+M3*M3+S*S);
     M1:=M1/DEN;
     M2:=M2/DEN;
     M3:=M3/DEN;
     S:=S/DEN;
     I:=I+1;
     GOIQ ITER1;

```



```

PHASE2:M1:=0;
M2:=0;
M3:=1;
S:=0;
I:=N;
ITER2:A2:=M2;
B2:=M3;
C2:=S-M1*Y(/I/);
DEN:=A1(/I/)*B2-A2*B1(/I/);
IF ABS(DEN)<=1-6 THEN
  BEGIN
    OUTSTRING(1,"IMPOSSIBLE");
    GOIQ TERM
  END ;
DY(/I/):=(C1(/I/)*B2-C2*B1(/I/))/DEN;
D2Y(/I/):=(A1(/I/)*C2-A2*C1(/I/))/DEN;
IF I=1 THEN GOIQ TERM1;
H:=T(/I/)-T(/I-1/);
F1:=M1;
F2:=M1*H+M2;
F3:=M1*H*H/2+M2*H+M3;
F4:=M1*H*H*H/6+M2*H*H/2+M3*H;
P:=H*H*H/6;
M1:=F1*P-F4;
M2:=F2*P-F4*H;
M3:=F3*P-F4*H*H/2;
S:=S*P-F4*Y(/I/);
DEN:=SQRT(M1*M1+M2*M2+M3*M3+S*S);
M1:=M1/DEN;
M2:=M2/DEN;
M3:=M3/DEN;
S:=S/DEN;
I:=I-1;
GOIQ ITER2;
TERM1: FOR I:=1 STEP 1 UNTIL N-1 DO
  BEGIN
    H:=T(/I+1/)-T(/I/);
    D3YP(/I/):=6*((Y(/I+1/)-Y(/I/))/(H*H*H)-DY(/I/)/(H*H)-D2Y(/I/)/(2*H));
  END ;
D3YP(/N/):=0;
D3YM(/1/):=0;
FOR I:=2 STEP 1 UNTIL N DO
  BEGIN
    H:=T(/I-1/)-T(/I/);
    D3YM(/I/):=6*((Y(/I-1/)-Y(/I/))/(H*H*H)-DY(/I/)/(H*H)-D2Y(/I/)/(2*H));
  END ;
FOR I:=1 STEP 1 UNTIL N DO MU(/I/):=D3YP(/I/)-D3YM(/I/);
TERM:
END ;

```


6)

```

REAL PROCEDURE DIFDIV(J,L,X,K);
VALUE J,L,X,K;
INTEGER J,K;
ARRAY X,L;

```

```

BEGIN
  INTEGER I,R;
  FOR R:=1 STEP 1 UNTIL K DO FOR I:=0 STEP 1 UNTIL K-R
    DO L(I/I):=(L(I/I+1)-L(I/I))/(X(I/J+I+R)-X(I/J+I));
  DIFDIV:=L(0/0)
END DIF DIV;

```

```

REAL PROCEDURE FI(I,T,X,K);
VALUE I,T,X,K;
ARRAY X;
INTEGER K,I;
REAL T;

```

```

BEGIN
  ARRAY L(0:K/);
  INTEGER P,R;
  IF T>X(I+K/) THEN FI:=0 ELSE
  BEGIN
    FOR R:=0 STEP 1 UNTIL K DO L(R/):= IF X(I+R/)>T
      THEN (X(I+R/)-T)**(2*K-1) ELSE 0;
    FOR R:=1 STEP 1 UNTIL K DO FOR P:=0 STEP 1
      UNTIL K-R DO L(P/):=(L(P+1/)-L(P/))/(X(I+P+R/)-X(I+P/));
    FI:=L(0/0)
  END
END FI;

```

```

PROCEDURE CONSTR(G,K,N,X);
ARRAY G,X;
INTEGER K,N;

```

```

BEGIN
  INTEGER I,J,R;
  ARRAY L(0:K/);
  FOR I:=1 STEP 1 UNTIL N-K DO
  BEGIN
    FOR J:=1 STEP 1 UNTIL IE I+K-1<N-K THEN I+K-1
      ELSE N-K DO
    BEGIN
      FOR R:=0 STEP 1 UNTIL K DO L(R/):=FI(I,X(I/J+R/),X,K);
      G(I,J,I):=G(I,I,J):=DIFDIV(J,L,X,K)
    END;
    FOR J:=I+K STEP 1 UNTIL N-K DO G(I,J,I):=G(I,I,J):=0;
  END
END CONSTR;

```



```

PROCEDURE SAUDER IV(K,N,X,Y,G,MU);
INTEGER K,N;
ARRAY X,Y,G,MU;

```

```

BEGIN
  ARRAY LAMBDA,H(/1:N-K/),L(/0:K/),B(/1:N/);
  INTEGER I,J,M,R;
  REAL S,T;
  CCNSTRUCTIONDEB: FOR I:=1 STEP 1 UNTIL N-K DO

```

```

  BEGIN
    FOR R:=0 STEP 1 UNTIL K DO L(/R/):=Y(/I+R/);
    B(/I/):=DIFDIV(I,L,X,K);

```

```

  END ;
  CALL AMBD:GRESOL(G,B,LAMBDA,N-K,IMP);
  IMP:CALMU: FOR I:=1 STEP 1 UNTIL N DO

```

```

  BEGIN
    T:=0;
    FOR J:=1 STEP 1 UNTIL N-K DO IF J>1 OR J<I-K
    THEN H(/J/):=0 ELSE
      BEGIN
        S:=1;
        FOR R:=0 STEP 1 UNTIL I-J-1,I-J+1 STEP 1
          UNTIL K DO S:=S/(X(/I/)-X(/J+R/));
        H(/J/):=S

```

```

      END ;
      FOR J:=1 STEP 1 UNTIL N-K DO T:=T+H(/J/)*LAMBDA(/J/);
      M:=(-1)**K;
      FOR J:=2 STEP 1 UNTIL 2*K-1 DO M:=M*J;
      MU(/I/):=T*M

```

```

    END ;
  END SAUDER IV;

```

3)

```

PROCEDURE SAUBASE(X,SAU,N);
INTEGER N;
ARRAY X,SAU;

```

```

BEGIN
  INTEGER I,J;
  ARRAY Y,MU(/1:N/);
  FOR I:=1 STEP 1 UNTIL N DO
    BEGIN
      FOR J:=1 STEP 1 UNTIL I-1,I+1 STEP 1 UNTIL N DO Y(/J/):=0;
      Y(/I/):=1;
      SPLDEU(N,Y,MU);
      FOR J:=1 STEP 1 UNTIL N DO SAU(/I,J/):=MU(/J/)
    END
  END SAUBASE;

```


R E F E R E N C E S



- [1] P.M. ANSELONE A general method for the construction of interpolating
P.J. LAURENT or smoothing spline-functions.
Num.Math. 12, (1958), p.66-82.

- [2] M. ATTEIA Théorie et applications des fonctions "spline" en analyse
numérique.
Thèse, GRENOBLE (1966)

- [3] M. ATTEIA Fonctions "spline" avec contraintes linéaires de type
inégalité.
6ième Congrès de l'AFIRO, NANCY, Mai 1967.

- [4] M. ATTEIA Fonctions spline définies sur un ensemble convexe.
Num.Math. 12, (1968), p.192-210.

- [5] S. BANACH Opérations linéaires.
Chelsea publishing company -NEW-YORK (1955).

- [6] C.CARASSO Méthodes numériques pour l'obtention de fonctions-spline.
Thèse, GRENOBLE (1966).

- [7] C. CARASSO On the numerical constructions and the partical use of
P.J. LAURENT interpolating spline-functions.
IFIP Congress 68, EDINBURGH, Août 68, p.A6 - 10.

- [8] J. CEA Les méthodes de "descente" dans la théorie de l'optimisation.
Revue R.I.R.O n°13 R3-1968, p.79-102.
- [9] A.Ja.DUBOVICKII Problèmes d'optimisation en présence de contraintes.
A.A. MILJUTIN Th.Vychisl.Mat.I Mat.Fiz., 5, (1965), p.395-453.
- [10] H.G. EGGLESTON Convexity.
CAMBRIDGE, Univ.Press (1963).
- [11] N. GASTINEL Analyse numérique linéaire.
PARIS, Hermann (1966).
- [12] E.G. GILBERT An iterative procedure for computing the minimum of a quadratic form on a convex set.
J.SIAM Control, vol.4, n°1, (1966), p 61-79.
- [13] Y. HAUGAZEAU Sur la minimisation des formes quadratiques avec contraintes
C.R. Acad.Sci. PARIS, 264, (1967), 322-324.
- [14] Y. HAUGAZEAU Sur les inéquations variationnelles et la minimisation de fonctionnelles convexes.
Thèse, PARIS, 7 juin 1968.
- [15] P.J.LAURENT Théorèmes de caractérisation en approximation convexe.
Colloque "théorie de l'approximation des fonctions".
(Roumanie), 15-20 sept.1967.
Mathematica 10(33), 11, 1968), 95-111.

- [16] P.J. LAURENT Cours de théorie de l'approximation.
Fascicule 4, Université de GRENOBLE, 1968-69.
- [17] P.J. LAURENT Théorème de caractérisation d'une meilleure approximation
dans un espace normé et généralisation de l'algorithme de
Remes.
Num. Math 10, (1967), 190-208.
- [18] P.J. LAURENT Méthodes duales pour le calcul du minimum d'une fonction
convexe sur une intersection de convexes.
B. MARTINET Colloque d'optimisation, NICE, 30 juin-5juillet 1969.
- [19] P.J. LAURENT Construction of spline-functions in a convex set.
Colloque d'Analyse Numérique- Mai 1969- MADISON
Université de WISCONSIN.
- [20] J.L. LIONS Une méthode d'éclatement des opérateurs et des contraintes
R. TEMAN en calcul de variations.
C.R. Acad. Sci.PARIS ; 263, (1966), 563-565.
- [21] C. LOBRY Etude géométrique des problèmes d'optimisation en présence
de contraintes.
Thèse, GRENOBLE, 20 Juin 1967.
- [22] J.B. ROSEN The gradient projection method for non linear programming.
Part I. Linear constraints.
J. Soc. Indust.Appl. Math.; Vol.8, n°1, March (1960).
- [23] I.J.SCHOENBERG Spline interpolation and the highes direvatives.*
Proc. of the Nat.Acad.od Sciences; 51, n°1, (1967) 24-28.
- [24] G.ZOUTENDIJK Methods of feasible directions.A study in linear and non-
linear programming.
Elsevier publishing Company. AMSTERDAM (1960).

VU

Grenoble, le

Le Président de la Thèse

VU

Grenoble, le

Le Doyen de la Faculté des Sciences

Vu, et permis d'imprimer,

Le Recteur de l'Académie de GRENOBLE

