N° d'ordre : 508

50376
1975
102

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

DOCTEUR de SPECIALITE

(MATHEMATIQUES APPLIQUEES)

par

Jean BEUNEU

ADAPTATION DE LA METHODE DES CENTRES LINEARISEE AUX PROBLEMES DE GRANDE TAILLE

Soutenue le 21 février 1975, devant la COMMISSION D'EXAMEN

Membres du Jury :	MM.	P. BACCHUS	Président
		C. BREZINSKI	Examinateur
		P. HANSEN	Invité
		P. HUARD	Rapporteur

UNIVERSITE DES SCIENCES
ET TECHNIQUES DE LILLE

DOYENS HONORAIRES DE L'Ancienne Faculté des Sciences

MM. R. DEFRETIN, H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES Des Anciennes Facultés de Droit Et Sciences Economiques, des Sciences et des Lettres

M. ARNOULT, Mme BEAUJEU, MM. BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, P. GERMAIN, HEIM DE BALSAC, HOCQUETTE, KAMPE DE FERIET, KOUGANOFF, LAMOTTE, LASSERRE, LELONG, Mme LELONG, MM. LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, NORMANT, PEREZ, ROIG, ROSEAU, ROUBINE, ROUELLE, SAVART, WATERLOT, WIEMAN, ZAMANSKI.

PRESIDENT HONORAIRE DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

R. DEFRETIN

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

M. PARREAU.

PROFESSEURS TITULAIRES

M. BACCHUS Pierre	Astronomie
M. BEAUFILS Jean-Pierre	Chimie Physique
M. BECART Maurice	Physique Atomique et Moléculaire
M. BLAYS Pierre	Géographie
M. BONNEMAN Pierre	Chimie Appliquée
M. BONTE Antoine	Géologie Appliquée
M. BOUGHON Pierre	Algèbre
M. BOUISSET Simon	Physiologie Animale
M. BOURIQUET Robert	Biologie Végétale
M. CELET Paul	Géologie Générale
M. CONSTANT Eugène	Electronique
M. CORSIN Pierre	Paléontologie
M. DECUYPER Marcel	Géométrie
M. DELATTRE Charles	Géologie Générale
M. DELHAYE Michel	Chimie Physique
M. DERCOURT Michel	Géologie Générale
M. DURCHON Maurice	Biologie Expérimentale
M. FAURE Robert	Mécanique
M. FOURET René	Physique du Solide
M. GABILLARD Robert	Electronique
M. GLACET Charles	Chimie Organique
M. GONTIER Gérard	Mécanique
M. GRUSON Laurent	Algèbre
M. GUILLAUME Jean	Microbiologie
M. HEUBEL Joseph	Chimie Minérale
M. LANSRAUX Guy	Physique Atomique et Moléculaire
M. LEBRUN André	Electronique
M. LEHMANN Daniel	Géométrie
Mme LENOBLE Jacqueline	Physique Atomique et Moléculaire
M. LINDER Robert	Biologie et Physiologie Végétales
M. LOMBARD Jacques	Sociologie

.../...

M. LUCQUIN Michel	Chimie Physique
M. MAILLET Pierre	Sciences Economiques
M. MONTARIOL Frédéric	Chimie Appliquée
M. MONTREUIL Jean	Biochimie
M. PARREAU Michel	Analyse
M. POUZET Pierre	Analyse Numérique
M. PROUVOST Jean	Minéralogie
M. SCHILTZ René	Physique Atomique et Moléculaire
Mme SCHWARTZ Marie-Hélène	Géométrie
M. TILLIEU Jacques	Physique Théorique
M. TRIDOT Gabriel	Chimie Appliquée
M. VAILLANT Jean	Analyse
M. VIDAL Pierre	Automatique
M. VIVIER Emile	Biologie Cellulaire
M. WERTHEIMER Raymond	Physique Atomique et Moléculaire
M. ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS SANS CHAIRE

M. BELLET Jean	Physique Atomique et Moléculaire
M. BILLARD Jean	Physique du Solide
M. BODARD Marcel	Biologie Végétale
M. BOILLET Pierre	Physique Atomique et Moléculaire
M. BONNOT Ernest	Biologie Végétale
M. BRIDOUX Michel	Chimie Physique
M. CAPURON Alfred	Biologie Animale
M. DEPREZ Gilbert	Physique Théorique
M. DEVRAINNE Pierre	Chimie Minérale
M. GOUDMAND Pierre	Chimie Physique
M. GUILBAULT Pierre	Physiologie Animale
M. LABLACHE-COMBIER Alain	Chimie Organique
M. LACOSTE Louis	Biologie Végétale
Mme LEHMANN Josiane	Analyse
M. LOUCHEUX Claude	Chimie Physique
M. MAES Serge	Physique Atomique et Moléculaire
Melle MARQUET Simone	Probabilités
M. MIGEON Michel	Chimie Physique
M. MONTEL Marc	Physique du Solide
M. PANET Marius	Electrotechnique
M. RACZY Ladislas	Electronique
M. ROUSSEAU Jean-Paul	Physiologie Animale
M. SALMER Georges	Electronique
M. SEGUIER Guy	Electrotechnique

MAITRES DE CONFERENCES (et chargés d'Enseignement)

M. ADAM Michel	Sciences Economiques
M. ANDRE Charles	Sciences Economiques
M. ANGRAND Jean-Pierre	Géographie
M. ANTOINE Philippe	Analyse
M. BART André	Biologie Animale
M. BEGUIN Paul	Mécanique
M. BKOUCHE Rudolphe	Algèbre
M. BOILLY Bénoni	Biologie Animale
M. BONNEMAIN Jean-Louis	Biologie Végétale
M. BOSCQ Denis	Probabilités
M. BREZINSKI Claude	Analyse Numérique
M. BRUYELLE Pierre	Géographie
M. CARREZ Christian	Informatique
M. CORDONNIER Vincent	Informatique
M. CORTOIS Jean	Physique Nucléaire et Corpusculaire

.../...

M. COQUERY Jean-Marie	Psycho-Physiologie
M. COULON Jean	Electrotechnique
Mlle DACCHARI Monique	Géographie
M. DEBOURSE Jean-Pierre	Gestion des Entreprises
M. DEBRABANT Pierre	Géologie Appliquée
M. DHAINAUT André	Biologie Animale
M. DELAUNAY Jean-Claude	Sciences Economiques
M. DERIEUX Jean-Claude	Microbiologie
M. DOUKHAN Jean-Claude	Physique du Solide
M. DRIEUX Baudouin	Informatique
M. DUEE Gérard	Géologie Appliquée
M. DYMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du Solide
Mme EVRARD Micheline	Chimie Appliquée
M. FONTAINE Jacques-Marie	Electronique
M. FOURNET Bernard	Biochimie
M. FROELICH Daniel	Chimie Physique
M. GAMBLIN André	Géographie
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GRANELLE Jean Jacques	Sciences Economiques
M. GUIGOU Jean-Louis	Sciences Economiques
M. GUILLAUME Henri	Sciences Economiques
M. HECTOR Joseph	Géométrie
M. HERMAN Maurice	Physique Spatiale
M. JOURNEL Gérard	Physique Atomique et Moléculaire
Mlle KOSMANN Yvette	Géométrie
M. KREMBEL Jean	Biochimie
M. LANGRAND Claude	Probabilités
M. LAURENT François	Automatique
Mlle LEGRAND Denise	Algèbre
Mlle LEGRAND Solange	Algèbre
M. LENTACKER Firmin	Géographie
M. LEROY Jean-Marie	Chimie Appliquée
M. LEROY Yves	Electronique
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique Théorique
M. LOUAGE Francis	Electronique
M. MAHIEU Jean-Marie	Physique Atomique et Moléculaire
Mme N'GUYEN VAN CHI Régine	Géographie
M. MAIZIERES Christian	Automatique
M. MALAUSSENA Jean-Louis	Sciences Economiques
M. MESSELYN Jean	Physique Atomique et Moléculaire
M. MONTUELLE Bernard	Biologie Appliquée
M. NICOLE Jacques	Chimie Appliquée
M. PAQUET Jacques	Géologie Générale
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie Physique
M. PERROT Pierre	Chimie Appliquée
M. PERTUZON Emile	Physiologie Animale
M. PONSOLLE Louis	Chimie Physique
M. POVY Lucien	Automatique
M. ROGALSKI Marc	Analyse
M. ROY Jean-Claude	Psycho-Physiologie
M. SIMON Michel	Sociologie
M. SLIWA Henri	Chimie Organique
M. SOMME Jean	Géographie
Mlle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. THERY Pierre	Electronique
M. TOULOTTE Jean-Marc	Automatique
M. TREANTON Jean-René	Sociologie

.../...

M. VANDORPE Bernard	Chimie Minérale
M. VILLETTE Michel	Mécanique
M. WERNIER Georges	Informatique
M. WATERLOT Michel	Géologie Générale
M. YVON Jean-Pierre	Analyse Numérique
Mme ZINN JUSTIN Nicole	Algèbre

Je tiens à exprimer ma profonde gratitude à Monsieur le Professeur BACCHUS pour l'honneur qu'il me fait de présider le Jury.

Je remercie très vivement Monsieur le Professeur BREZINSKI et Monsieur le Professeur HANSEN qui ont bien voulu s'intéresser à ce travail et ont accepté de le juger.

Que Monsieur le Professeur HUARD, qui m'a donné l'idée de ce travail et m'a permis de le mener à bien par ses précieux conseils et suggestions, veuille bien trouver ici l'expression de ma plus vive reconnaissance.

Que soient également remerciés les membres du Groupe de Recherche Opérationnelle du Laboratoire de Calcul de l'U.E.R d'I.E.E.A pour l'amical intérêt qu'ils n'ont cessé de me témoigner.

Je tiens enfin à remercier Mademoiselle DESQUIENS ainsi que Monsieur et Madame DEBOCK qui, avec gentillesse, compétence et efficacité, ont assuré la réalisation matérielle de cette thèse.

TABLE DES MATIERES

INTRODUCTION

La méthode des centres est un algorithme général de résolution des programmes mathématiques non linéaires. Cette méthode est introduite dans [5] par HUARD, qui en propose dans [22] une importante variante, la méthode des centres linéarisée ; cette variante utilise la linéarisation des fonctions du problème considéré.

DENEL applique dans [10] la méthode des centres linéarisée à la résolution des problèmes-tests de l'étude de COLVILLE ([8]) et obtient des résultats intéressants en utilisant l'idée classique d'adjonction de contraintes additionnelles pour accélérer la convergence pratique de la méthode (algorithme de centrage). Ces problèmes-tests sont d'origine et de forme diverses, mais ont pour caractéristique commune essentielle leur taille réduite (16 variables ou 14 contraintes au maximum).

Or, les exemples concrets de programmes mathématiques comportent fréquemment quelques centaines de variables et de contraintes, et cette taille importante introduit de nouvelles difficultés : l'efficacité de l'algorithme de centrage est diminuée, l'encombrement mémoire croît rapidement et les temps de calcul peuvent être longs.

En outre, les problèmes concrets comportent fréquemment des contraintes d'égalités qui empêchent à priori leur résolution par la méthode des centres linéarisée. Enfin, on ne connaît pas en général, de point de départ de l'algorithme.

Le but de ce travail est d'adapter la méthode des centres linéarisée de manière à résoudre ces difficultés.

Après avoir rappelé les caractéristiques générales de la méthode des centres et de la méthode des centres linéarisée, nous exposons au chapitre I une généralisation de l'algorithme de centrage ainsi qu'une variante plus efficace de cet algorithme. Puis nous donnons un procédé simple d'obtention d'un point de départ.

Nous dégageons ensuite les aspects numériques de la méthode ; à chaque itération interviennent deux problèmes qui représentent l'essentiel des calculs : résolution de programmes linéaires et maximisation de fonctions non continuement différentiables sur un segment de $\mathbb{R}^n$.

Le premier de ces problèmes est abordé au chapitre II où nous exposons une technique, liée à la méthode simpliciale, efficace du point de vue de l'encombrement mémoire, et pouvant conduire à une réduction du volume des calculs.

Nous envisageons ensuite une autre technique liée aux méthodes de relaxation.

Le second de ces problèmes, maximisation d'une fonction non continuement différentiable sur un segment (cette fonction changeant à chaque itération) est abordé au chapitre III. Nous rappelons divers procédés classiques et nous introduisons une technique d'interpolation polygonale bien adaptée à la forme de la fonction.

Nous montrons au chapitre IV comment il est possible de prendre en compte les problèmes comportant des contraintes d'égalités en transformant ces contraintes en inéquations.

A titre d'application, nous considérons au chapitre V un problème d'origine concrète pouvant être de taille importante (distribution d'énergie électrique), et nous terminons au chapitre VI par des expériences numériques systématiques portant sur 6 problèmes de cette forme, de taille allant jusqu'à 103 variables et 88 contraintes, qui ont été résolus complètement avec une grande précision.

NOTATIONS

$\mathbb{N}$ ensemble des nombres entiers

$\mathbb{R}$ ensemble des nombres réels

$\mathbb{R}^+$ ensemble des nombres réels strictement positifs

$\forall A \subset \mathbb{R}^n$:

$\mathring{A}$ intérieur de A

Fr (A) frontière de A

$\forall x \in \mathbb{R}^n$:

$||x||$ norme euclidienne de x.

I et J étant deux ensembles finis d'indices :

$|I|$ nombre d'éléments de I

La matrice $M : I \times J \to \mathbb{R}$ est indicée en lignes par I et en colonnes par J.

M_i^j élément (i,j) de M

M_i restriction de M à $\{i\} \times J$ (ligne i de M)

M^j restriction de M à $I \times \{j\}$ (colonne j de M)

Si $I' \subset I$ et $J' \subset J$:

$M_{I'}^{J'}$ restriction de M à $I' \times J'$ (sous-matrice$(I' \times J')$ de M).

M^t transposée de M ($(M^t)_i^j = M_j^i$ $\forall i \in J$, $\forall j \in I$).

$\forall f : \mathbb{R}^n \to \mathbb{R}$, différentiable en $x^* \in \mathbb{R}$:

$\nabla f(x^*)$ vecteur ligne représentant le gradient de f en x^*.

$\forall f : \mathbb{R}^n \to \mathbb{R}$, deux fois différentiable en $x^* \in \mathbb{R}$:

$\nabla^2 f(x^*)$ hessien de f en x^* : c'est une matrice carrée symétrique de terme général

$$\frac{\partial^2 f(x^*)}{\partial x_i \, \partial x_j}$$

CHAPITRE I

METHODE DES CENTRES
ET
METHODE DES CENTRES LINEARISEE

I.1 Introduction :

Nous considérons le problème général de l'optimisation non linéaire, que l'on peut formuler de la façon suivante :

$$\left\| \begin{array}{l} \text{Maximiser } f(x) \\ \text{sous les conditions} \\ x \in A \subset \mathbb{R}^n. \end{array} \right.$$

La résolution de ce problème consiste à déterminer un point $\hat{x}$ du domaine des solutions réalisables A, vérifiant la condition d'optimalité globale:

$$f(x) \leqslant f(\hat{x}) \quad \forall x \in A.$$

On rencontre fréquemment dans la pratique des problèmes pouvant être modélisés sous cette forme ; on est donc conduit à introduire des algorithmes de résolution.

La méthode des Centres, proposée par HUARD ([5], [21]), est une méthode de résolution de tels problèmes ; il s'agit d'un schéma très général, auquel peuvent se rattacher de nombreuses méthodes connues.

Nous exposons dans ce chapitre les principales caractéristiques de la méthode, ainsi qu'une variante importante, la méthode des Centres linéarisée. Nous ne donnons ici aucune démonstration des propriétés citées, préférant renvoyer le lecteur aux articles de base de HUARD ([5], [21], [22] et [23]).

I.2 La méthode des Centres :

I.2.1 *Généralités*.

La méthode est un algorithme itératif qui construit une suite de points appartenant à des sous-ensembles du domaine des solutions réalisables appelés "tronçons", tels que chacun d'eux soit

"éloigné le plus possible" de la frontière du tronçon; ce sont donc des "centres géométriques", d'où le nom de centre donné à ces points. Cet éloignement se mesure à l'aide de fonctions particulières appelées F-distances.

I.2.2 *Problème envisagé.*

On considére le Programme mathématique

$$(P) \quad \left\| \begin{array}{l} \text{Maximiser } f(x) \\ x \in A \cap B \end{array} \right. .$$

avec $A \subset \mathbb{R}^n$, tel que $\overset{\circ}{A} \neq \emptyset$

$B \subset \mathbb{R}^n$

et $A \cap B$ fermé.

$f : \mathbb{R}^n \to \mathbb{R}$ continue ; on suppose que cette fonction atteint son maximum sur $A \cap B$ en un point $\hat{x}$, et que

$\mathrm{Fr}\{x \mid f(x) \geq \lambda\} = \{x \mid f(x) = \lambda\}$ $\forall \lambda \in \mathbb{R}$, $\lambda < f(\hat{x})$.

On fait de plus l'hypothèse suivante :

$\overset{\circ}{A} \cap B \cap O = \emptyset \Longrightarrow A \cap B \cap O = \emptyset$ $\forall O$ ouvert $\subset \mathbb{R}^n$.

Cette dernière propriété est vérifiée, en particulier, si $\overset{\circ}{A} \cap B \neq \emptyset$ et $A \cap B$ convexe.

I.2.3 *Notion de F-distance, d'ε-centre et de centre.*

Soit $E \subset P(\mathbb{R}^n)$ un ensemble de parties de $\mathbb{R}^n$.

Définition I.1 :

On appelle F-distance une fonction $d : \mathbb{R}^n \times E \to \mathbb{R}$ telle que :

1) $d(x,E) = o$, $\forall E \in E$, $\forall x \in \mathrm{Fr}(E)$

2) $d(x,E) > o$, $\forall E \in E$, $\forall x \in \overset{\circ}{E}$

3) $\forall E \in E$, $\forall E' \in E$, $E \subset E$; $\forall x \in E$, $\exists \rho(x) \in \mathbb{R}^+$

tel que $d(x,E) \leq \rho(x) d(x,E')$.

Définition I.2 :

Une F-distance définie sur $\mathbb{R}^n \times E$ est dite <u>régulière</u> si :

$\forall \{E_k \in E | k \in \mathbb{N}\}$ et $\forall \{\overset{k}{x} \in \mathbb{R}^n | k \in \mathbb{N}\}$

tels que

$E_k \supset E_{k+1} \supset E \in E$, $E \neq \emptyset$

$\forall k$, $\overset{k}{x} \in E_k$, $\overset{k}{x} \notin \overset{\circ}{E}_{k+1}$

on a $d(\overset{k}{x}, E_k) \longrightarrow 0$ quand $k \longrightarrow \infty$.

Définition I.3 :

Etant donnés $E \in E$, une F-distance d sur $\mathbb{R}^n \times E$ et un nombre $\varepsilon \in \mathbb{R}^+$ tel que $0 \leq \varepsilon < \sup \{d(x,E) | x \in E\}$, on appelle <u>$\varepsilon$-centre</u> de E relativement à d tout point $c \in E$

tel que

$$d(c,E) \geq \sup \{d(x,E) | x \in E\} - \varepsilon$$

Si $\varepsilon = o$, le point c est appelé <u>centre</u>.

La recherche d'un ε-centre de E relativement à d revient à résoudre

$$(Q) \quad \left\| \begin{array}{l} \text{Maximiser } d(x,E) \text{ à } \varepsilon \text{ près} \\ x \in E \end{array} \right.$$

L'optimum étant une solution intérieure, ce problème se ramène, sur le plan pratique, à une maximisation sans contraintes.

Remarque :

La solution de ce problème n'est pas unique ; même dans le cas où $\varepsilon = o$ on aura en général plusieurs solutions.

I.2.4 *Choix d'un ensemble E pour le problème (P).*

On considère des troncatures de l'ensemble A par des équipotentielles de la fonction économique.

Soit $K \subset \mathbb{R}^n$, défini par :

$$K =]\min \{f(x) | x \in A \cap B\}, \max \{f(x) | x \in A \cap B\}[$$

On désigne par $E(\lambda)$ un tronçon de A :

$$E(\lambda) = \{x | x \in A, f(x) \geq \lambda\} \quad \forall \lambda \in K$$

et on définit $E = \{E(\lambda) | \lambda \in K\}$.
Il est clair que $\forall \lambda, \lambda' \in K$, $\lambda > \lambda'$, on a $E(\lambda) \subset E(\lambda')$

On peut donc définir sur $R^n \times E$ une F-distance d régulière.

I.2.5 *Algorithme de la méthode des Centres.*

On choisit une suite décroissante de nombres réels $\varepsilon_k \geq o$, tendant vers o quand $k \to \infty$.

On se donne, au départ

$$\lambda_o \in K.$$

Chaque itération k de la méthode consiste en la suite d'opérations suivante :

On dispose de

$$\overset{k}{x} \in A \cap B.$$

Soit

$$\lambda_k = f(\overset{k}{x}) \leq f(\hat{x}).$$

Considérons le tronçon

$$E(\lambda_k) = \{x \mid x \in A,\ f(x) \geq \lambda_k\}$$

Si $\overset{\circ}{E}(\lambda_k) \cap B = \emptyset$, l'optimum $\hat{x}$ est atteint.
Sinon, on détermine un ε_k-centre de $E(\lambda_k)$ "relativement à B", c'est-à-dire qu'on résout

$$(Q_k) \quad \left\| \begin{array}{l} \text{Maximiser } d(x, E(\lambda_k)) \text{ à } \varepsilon_k \text{ près} \\ x \in E(\lambda_k) \cap B \end{array} \right.$$

On trouve donc $\overset{k+1}{x}' \in E(\lambda_k) \cap B$
tel que

$$d(\overset{k+1}{x}', E(\lambda_k)) \geq \sup \{d(x, E(\lambda_k)) \mid x \in E(\lambda_k) \cap B\} - \varepsilon_k.$$

On pose $\lambda'_{k+1} = f(\overset{k+1}{x}')$ et on prend $\overset{k+1}{x} \in E(\lambda'_{k+1}) \cap B$
(par exemple $\overset{k+1}{x} = \overset{k+1}{x}'$).

Remarque :

les ε_k doivent être choisis tels que

$$0 \leq \varepsilon_k < \sup \{d(x, E(\lambda_k)) \mid x \in E(\lambda_k) \cap B\}.$$

pour assurer que $\overset{k+1}{x}'$ appartient à l'intérieur du tronçon $E(\lambda_k)$.

PROPOSITION I 1 : (convergence)

Si, à une étape de l'algorithme, on a $\overset{\circ}{E}(\lambda_k) \cap B = \emptyset$ alors $\overset{k}{x}$ est solution optimale de (P).

Sinon, la suite infinie $\{\overset{k}{x} | k \in \mathbb{N}\}$ est telle que,

$$\forall k \in \mathbb{N} : \overset{k}{x} \in A \cap B, f(\overset{k}{x}) < f(\overset{k+1}{x}) < f(\hat{x})$$

$$\{f(\overset{k}{x}) | k \in \mathbb{N}\} \longrightarrow f(\hat{x})$$

et tout point d'accumulation de $\{\overset{k}{x}\}$ est solution optimale de (P).

Démonstration : [5], [21]

I.3 **La linéarisation.**

I.3.1 *Particularisation du problème (P).*

Nous considérons dans la suite le problème (P) particularisé de la façon suivante :

Soient

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

$$g_i : \mathbb{R}^n \longrightarrow \mathbb{R}, \; i \in L = \{1,2,\ldots m\}$$

des fonctions concaves et continuement différentiables.

On définit l'ensemble A par

$$A = \{x \in \mathbb{R}^n | g_i(x) \geq 0, \forall i \in L\}.$$

A est donc un ensemble convexe fermé de $\mathbb{R}^n$.

On prend pour B un ensemble convexe fermé de $\mathbb{R}^n$.

Les hypothèses de I.2.2 sont alors vérifiées ([5],[21]).

I.3.2 *Linéarisation* :

Les procédés de linéarisation consistent à ramener la résolution du problème non linéaire (P) à celle d'une suite de programmes linéaires.

Pour cela, étant donnés une fonction non linéaire différentiable

$$h : \mathbb{R}^n \longrightarrow \mathbb{R};$$

et un point

$$y \in \mathbb{R}^n$$

dit "point de linéarisation", on substitue à h la fonction

$$h' : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$$

telle que

$$h'(x;y) = h(y) + \nabla h(y).(x - y) \quad \forall x \in \mathbb{R}^n.$$

h' est affine par rapport à x et h'(.,y), où y est fixé, est appelée la "linéarisation de h au point y".

La valeur des fonctions h et h' sont égales si x = y.
La surface équipotentielle h'(x;y) = h'(y;y) est donc un hyperplan tangent à l'équipotentielle h(x) = h(y), d'où le nom d'approximation tangentielle donné à ce procédé.

On voit qu'une telle approximation ne sera en général valable qu'au voisinage de y, en particulier dans les cas de courbure forte de h(x).

PROPOSITION I.2 :

Si h est concave, on a, pour y fixé

$$h(x) \leq h'(x;y) \quad \forall x,y \in \mathbb{R}^n.$$

Ceci est une conséquence immédiate des propriétés classiques sur la convexité des fonctions.

L'algorithme itératif qui consiste, à chaque étape k, à résoudre un problème obtenu par linéarisation des fonctions f et g_i de (P) en prenant pour point de linéarisation l'optimum du problème de l'étape k - 1, ne converge pas, comme il est montré dans [24].

On trouve dans la littérature plusieurs méthodes basées sur la linéarisation ; elles peuvent se rattacher à l'une des trois approches suivantes :

1) Linéarisation des fonctions en un grand nombre de points choisis arbitrairement (méthodes à grille).

2) Linéarisation des fonctions en des points déterminés de manière itérative, les linéarisations précédentes étant conservées à chaque itération (méthodes de plans sécants).

3) Linéarisation des fonctions en des points déterminés de manière itérative, mais sans conserver les linéarisations précédentes (méthodes de directions améliorantes).

La méthode des Centres linéarisée, que nous allons maintenant envisager, se rattache à la 3e approche. Toutefois, le procédé d'accélération de convergence, dit "algorithme de centrage", que nous exposerons ensuite, se rattache à la 2e approche.

I.4. La méthode des Centres par linéarisations.

(appelée communément, par abus de langage, méthode des Centres linéarisée).

I.4.1 *Généralités :*

Dans l'algorithme de la méthode des Centres (I.2.5) nous avons vu que l'essentiel des calculs à chaque itération k consiste à résoudre (Q_k), c'est-à-dire maximiser sans contrainte, à ε_k près, la F-distance sur le tronçon $E(\lambda_k) \cap B$.

Le principe de la méthode des Centres linéarisée est de résoudre (Q_k) par une suite, finie ou infinie, de programmes linéaires ayant le même nombre de contraintes que (P).

On suppose que B est un polyèdre linéaire fermé borné de $\mathbb{R}^n$, donc convexe.

On définit sur $\mathbb{R}^n \times E$ la fonction :

$$d(x,E(\lambda)) = \min \{k_f(f(x) - \lambda), g_i(x) | i \in L\}. \quad k_f \in \mathbb{R}^+.$$

PROPOSITION I.3 :

La fonction d est une F-distance régulière sur $\mathbb{R}^n \times E$.

Démonstration : [23]

I.4.2 *Recherche d'un ε-centre du tronçon $E(\lambda) \cap B$.*

Soient $g'_i : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$, $\forall i \in L$, des fonctions numériques définies par :

$$g'_i(x;y) = g_i(y) + \nabla g_i(y).(x - y)$$

et soit $f' : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ la fonction numérique définie par :

$$f'(x;y) = f(y) + \nabla f(y).(x - y)$$

Définition I.4. :

On appelle F-distance linéarisée la fonction

$$d' : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \longrightarrow \mathbb{R}$$

telle que

$$d'(x,\lambda;y) = \min \{k_f(f'(x;y)-\lambda),\ g'_i(x;y) \mid i \in L\}.$$

PROPOSITION I.4. :

Pour y fixé, la F-distance linéarisée est une fonction affine par morceaux, continue et concave.
De plus, en utilisant la concavité des fonctions f et g_i, on a :

$$d'(x,\lambda;y) \geq d(x,\lambda) \quad \forall x,y \in \mathbb{R}^n, \quad \forall \lambda \in \mathbb{R}.$$

Démonstration : [22]

La 2e propriété est une conséquence immédiate de la proposition I.2

ALGORITHME PARTIEL.

1) Choisir un point de départ $\overset{o}{y} \in B$.
   Faire h = o

2) Résoudre le programme mathématique

$$Q'(\lambda;\overset{h}{y}) \quad \left\| \begin{array}{l} \text{Maximiser } d'(x,\lambda;\overset{h}{y}) \\ x \in B \end{array} \right.$$

   Soit $\overset{h}{z}$ une solution optimale de $Q'(\lambda;\overset{h}{y})$.

3) Déterminer $\overset{h+1}{y}$: $d(\overset{h+1}{y}, E(\lambda)) = \max \{d(x,E(\lambda)) \mid x \in [\overset{h}{y},\overset{h}{z}]\}$

4) Faire h = h+1 et aller en 2).

On obtient ainsi une suite infinie de points $\overset{h}{y}$, tels que $d(\overset{h}{y},E(\lambda))$ est une suite monotone non décroissante.

PROPOSITION I.5. :

La suite $\{\overset{h}{y}\}$ converge vers un centre du tronçon $E(\lambda)\cap B$.

Démonstration : [22]

Remarque 1 :

L'algorithme partiel impose $\overset{\circ}{y} \in B$, mais non nécessairement $\overset{\circ}{y} \in A$. Ceci est dû aux propriétés particulières de la F-distance d choisie. En pratique cependant, l'algorithme partiel étant utilisé pour trouver un centre du tronçon $E(\lambda_k)\cap B$, on prendra

$$\overset{\circ}{y} = \overset{k}{x}.$$

Remarque 2 :

Si on remplace le point 3) de l'algorithme partiel par

3') Déterminer $\overset{h+1}{y}$ tel que

$$d(\overset{h+1}{y},E(\lambda)) \geq d(\overset{h}{y},E(\lambda)) + \rho[\max\{d(x,E(\lambda)) \mid x \in [\overset{h}{y},\overset{h}{z}]\} - d(\overset{h}{y},E(\lambda))]$$

avec $\rho \in]0,1]$ constante indépendante de h, l'algorithme partiel converge encore ([22]). Ce résultat est intéressant sur le plan pratique ; il permet en effet de n'effectuer la maximisation de la F-distance sur le segment $[\overset{h}{y},\overset{h}{z}]$ que de manière approchée, ce qui autorise les procédés de maximisation approchée utilisés par le calcul sur ordinateur.

I.4.3 *Utilisation de l'algorithme partiel.*

Dans l'algorithme de la méthode des Centres on utilise l'algorithme partiel pour déterminer, à chaque étape k, un ε_k-centre du tronçon $E(\lambda_k) \cap B$. En pratique, on ne calculera qu'un nombre fini de points $\overset{h}{y}$, en s'appuyant sur les propositions suivantes.

PROPOSITION I.6. :

Si dans l'algorithme partiel on s'arrête à la première solution $\overset{1}{y}$ trouvée (c'est-à-dire si on n'effectue qu'une seule linéarisation par troncature), et qu'on pose

$$\overset{k+1}{x} = \overset{1}{y}$$

pour passer à l'étape suivante, et si $\overset{o}{x} \in B \cap E(\lambda_o)$, on obtient encore une méthode de centres. En effet, $\overset{1}{y}$ est un ε_k-centre de $E(\lambda_k)$ relativement à B, c'est-à-dire que $\overset{k}{x} \in B \cap E(\lambda_k)$ $\forall k$ et $\{\varepsilon_k\} \longrightarrow o$ quand $k \to \infty$ avec

$$\varepsilon_k = \sup \{d(x,E(\lambda_k)) \mid x \in B \cap E(\lambda_k)\} - d(\overset{k}{x},E(\lambda_k)).$$

Démonstration : [22]

L'algorithme général s'écrit alors :

On se donne au départ $\lambda_o \in K$ et $\overset{o}{x} \in B \cap E(\lambda_o)$
Itération k :
On dispose de $\overset{k}{x} \in B \cap E(\lambda_k)$.
Soit

$$\lambda_k = f(\overset{k}{x}).$$

1) Déterminer $\overset{k}{z}$: $d'(\overset{k}{z},\lambda_k;\overset{k}{x}) = \max \{d'(x,\lambda_k;\overset{k}{x}) \mid x \in B\}$.

2) Déterminer $\overset{k+1}{x}$: $d(\overset{k+1}{x}, E(\lambda_k)) = \max \{d(x,E(\lambda_k)) \mid x \in [\overset{k}{x},\overset{k}{z}]\}$.

PROPOSITION I.7. :

Si de plus, au lieu de prendre $\overset{k+1}{x} = \overset{1}{y}$, on prend, comme dans l'algorithme général, $\overset{k+1}{x}' = \overset{1}{y}$ et $\overset{k+1}{x}$ dans le tronçon $E(\lambda'_{k+1}) \cap B$, on obtient encore une méthode de centres.

Démonstration : [23],

L'algorithme général s'écrit alors :

On se donne au départ $\lambda_o \in K$ et $\overset{o}{x} \in B \cap E(\lambda_o)$

Itération k :

On dispose de $\overset{k}{x} \in B \cap E(\lambda_k)$.

Soit

$$\lambda_k = f(\overset{k}{x}).$$

1) Déterminer $\overset{k}{z}$: $d'(\overset{k}{z}, \lambda_k; \overset{k}{x}) = \max \{d'(x, \lambda_k; \overset{k}{x}) \mid x \in B\}$.

2) Déterminer $\overset{k+1}{x}'$: $d(\overset{k+1}{x}', E(\lambda_k)) = \max\{d(x, E(\lambda_k)) \mid x \in [\overset{k}{x}, \overset{k}{z}]\}$

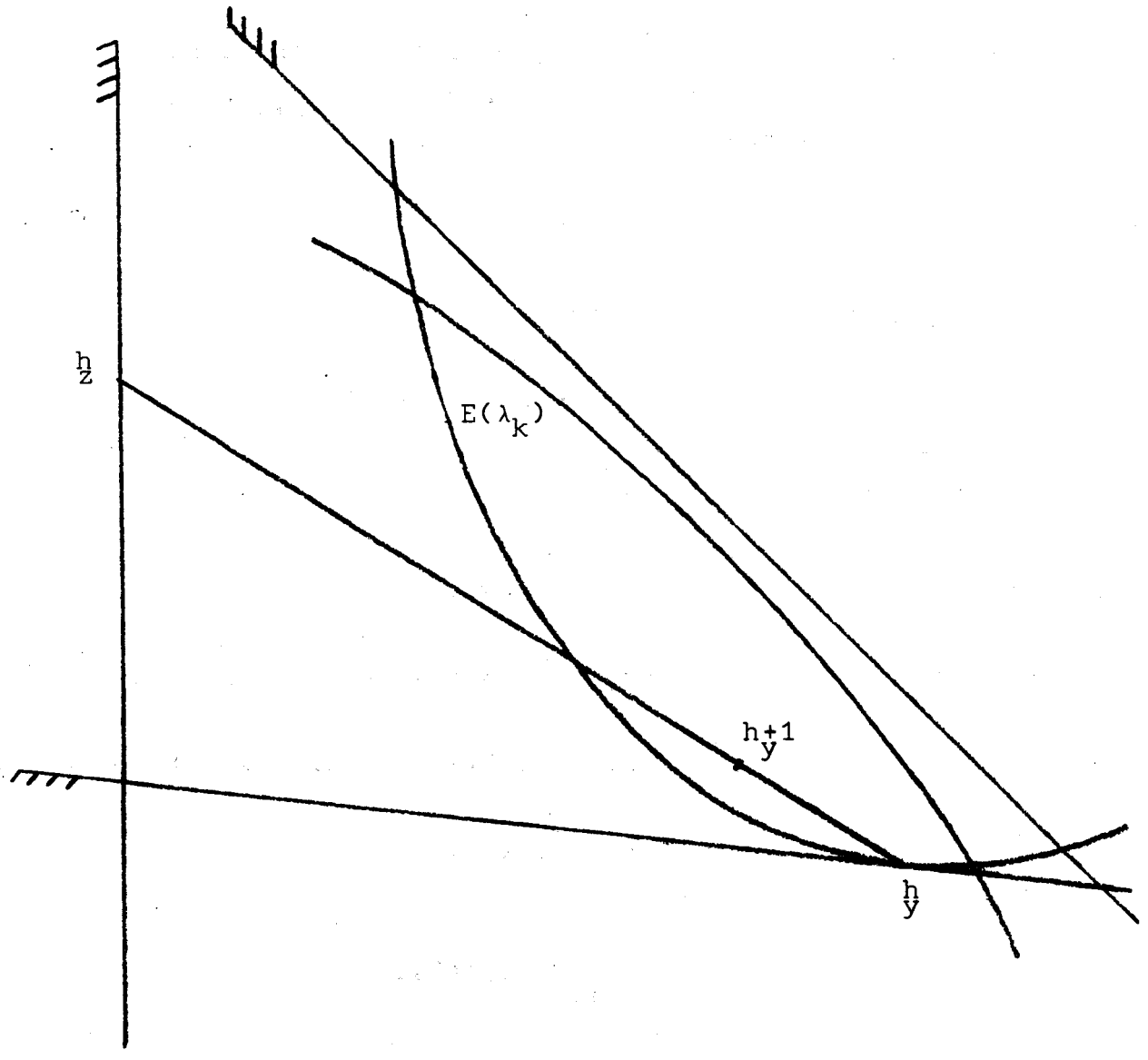
3) Choisir $\overset{k+1}{x}$ dans $E(\lambda'_{k+1}) \cap B$,
avec

$$\lambda'_{k+1} = f(\overset{k+1}{x}').$$

I.5. Algorithme de Centrage.

I.5.1 *Généralités.*

Dans bien des cas, la linéarisation des fonctions de (P) donne un polyèdre linéaire peu représentatif du tronçon $E(\lambda_k)$; l'approximation de $E(\lambda_k)$ par sa linéarisation étant mauvaise, les directions $[\overset{h}{y}, \overset{h}{z}]$ déterminées pénètrent très peu dans le tronçon, et l'ε_k-centre trouvé peut être très proche de la surface équipotentielle de la fonction économique, ce qui aboutit finalement à une faible amélioration sur la valeur de f(x).



La convergence pratique de la méthode peut donc être ralentie. Pour accélérer cette convergence pratique, on cherche à représenter plus efficacement le domaine non linéaire.

On se place dans le cadre de l'algorithme de la proposition I.7. L'algorithme de centrage réalise le point 3) de cet algorithme :
Choisir

$$x^{k+1}$$

dans

$$E(\lambda'_{k+1}) \cap B.$$

Il utilise la construction de contraintes additionnelles obtenues par linéarisation d'un contrainte ou de l'équipotentielle de la fonction économique en des points sur la frontière du tronçon.

I.5.2 *Algorithme de Centrage généralisé.*

Prendre

$$\overset{o}{t} = x^{k+1}, \quad \overset{o}{u} = z^{k}$$

Poser

$$d'o(x,\lambda_k;\overset{k}{x}) = d'(x,\lambda_k;\overset{k}{x}) = \min \{k_f(f'(x;\overset{k}{x})-\lambda_k), g'_i(x;\overset{k}{x}) \mid i \in L\} \quad \forall x \in \mathbb{R}^n.$$

Choisir un point

$$\overset{o}{v} \in E(\lambda_k) \cap B.$$

Faire $h = o$

1) Trouver, s'il existe, le point $\overset{h}{w}$ où le segment $[\overset{h}{v},\overset{h}{u}]$ rencontre la frontière du tronçon $E(\lambda_k)$:

$$\overset{h}{w} \in Fr(E(\lambda_k)) \cap [\overset{h}{v},\overset{h}{u}].$$

S'il n'existe pas de $\overset{h}{w}$, c'est-à-dire si $\overset{h}{u} \in E(\lambda_k)$, aller en 4).

Rechercher la contrainte passant par $\overset{h}{w}$; soit l(x) cette contrainte.

Définir une nouvelle F-distance linéarisée :

$$d'_{h+1}(x,\lambda_k;\overset{k}{x}) = \min \{d'_h(x,\lambda_k;\overset{k}{x}),\ l'(x;\overset{h}{w})\}.$$

2) Trouver $\overset{h+1}{u}$, optimum du programme mathématique

$$\left\| \begin{array}{l} \text{Maximiser } d'_{h+1}(x,\lambda_k;\overset{k}{x}) \\ x \in B \end{array} \right.$$

Déterminer $\overset{h+1}{t}$: $d(\overset{h+1}{t}, E(\lambda_k)) = \max\{d(x,E(\lambda_k)) \mid x \in [\overset{h}{v},\overset{h+1}{u}]\}$

Choisir un point

$$\overset{h+1}{v} \in E(\lambda_k) \cap B.$$

3) Si h = h*, h* étant le nombre maximum de contraintes additionnelles fixé à l'avance, aller en 4).

Sinon, faire h = h + 1 et aller en 1).

4) Choisir, parmi les points $\overset{\circ}{t},\overset{1}{t},\ldots\overset{h}{t}$, le point $\overset{h+1}{x}{}''$ qui donne la meilleure valeur de d.

Choisir pour point $\overset{k+1}{x}$ celui qui donne, parmi $\overset{k+1}{x}{}'$ ou $\overset{k+1}{x}{}''$, la meilleure valeur de la fonction économique f.

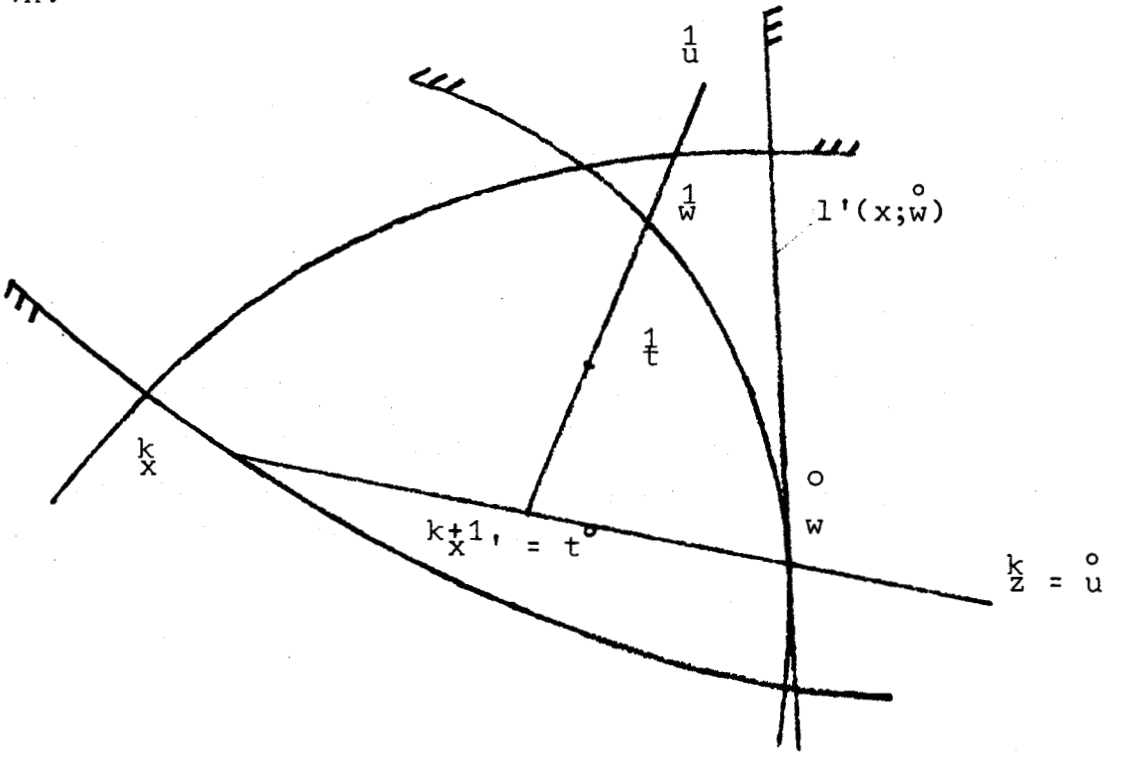
Cet algorithme généralise celui de [10] et [11].
A chaque étape, on ajoute donc une contrainte à l'expression de la F-distance linéarisée. Chaque contrainte additionnelle introduite tronque le polyèdre linéarisé de manière que l'optimum $\overset{h}{u}$ du programme précédent devienne extérieur au nouveau polyèdre. On est donc assuré que $\overset{h+1}{u} \neq \overset{h}{u}$.
D'autre part, pour ne pas trop augmenter le nombre de contraintes additionnelles et donc le volume des calculs, et pour éviter certains inconvénients numériques classiques dans les méthodes de linéarisation, dus au fait que la matrice des contraintes a tendance à devenir singulière, on donnera en pratique à h* une valeur peu élevée.

I.5.3 *Particularisation de l'algorithme.*

Le choix des points $\overset{h}{v}$ dans $E(\lambda_k) \cap B$ est en principe arbitraire.
En pratique, on a intérêt à les choisir de manière à obtenir la meilleure augmentation possible de d pour accélérer la convergence de la méthode.

CHOIX N°1 ([10],[11]).

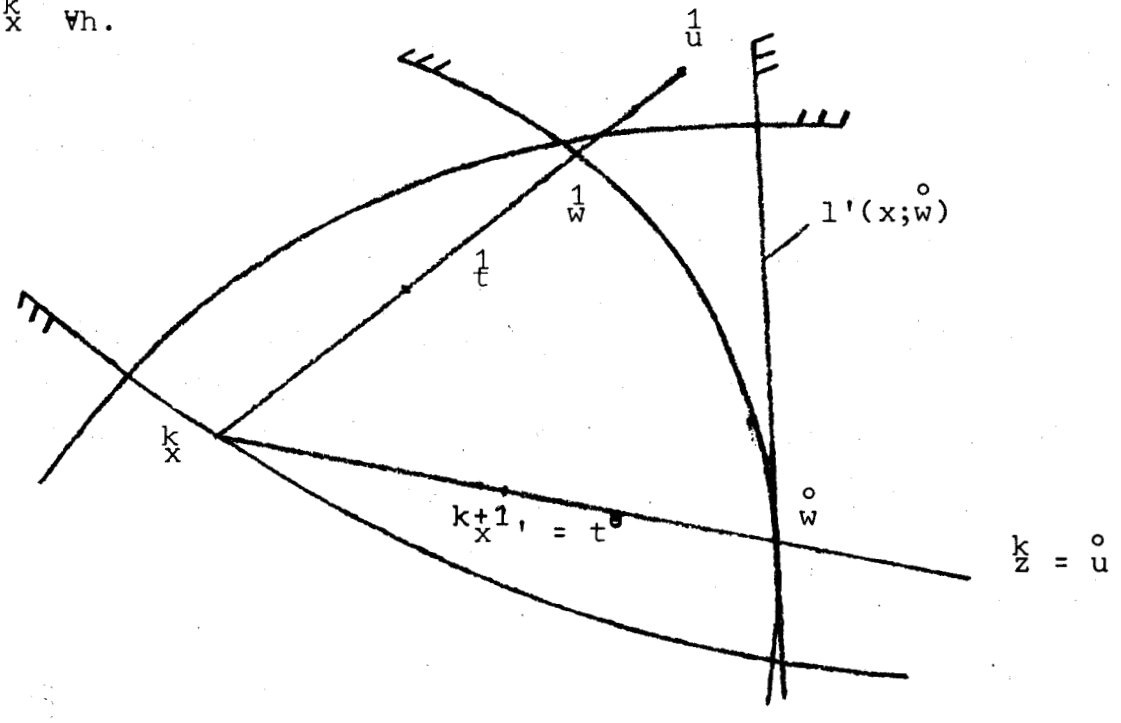
On prend $\overset{h}{v} = \overset{h}{t} \quad \forall h$.



Dans ce cas, on maximise toujours la F-distance d sur un segment partant du dernier point $\overset{h}{t}$ obtenu ; on est donc certain que la suite des valeurs de $d(\overset{h}{t}, E(\lambda_k))$ obtenue est non décroissante. Donc $\overset{k+1}{x}{}''$ = $\overset{h}{t}$ dans le point 4) de l'algorithme.

CHOIX N°2 ([2]).

On prend $\overset{h}{v} = \overset{k}{x}$ $\forall h$.



Dans ce cas, le segment considéré a toujours pour extrémité le même point $\overset{k}{x}$; la suite des valeurs de $d(\overset{h}{t},E(\lambda_k))$ obtenue n'est en général pas croissante.

Remarque :

Dans toutes les expériences numériques effectuées, aussi bien avec le choix n°1 qu'avec le choix n°2, l'algorithme de centrage a toujours donné un point $\overset{k+1}{x}{}''$ meilleur que $\overset{k+1}{x}{}'$ en valeur économique.
En pratique on a donc toujours pris $\overset{k+1}{x} = \overset{k+1}{x}{}''$.

I.6. Aspects complémentaires de la méthode.

I.6.1 *Pondération des fonctions*.

Nous avons défini (I.4.1) la F-distance par

$$d(x,E(\lambda)) = \min \{k_f(f(x) - \lambda), g_i(x) | i \in L\}, \; k_f \in \mathbb{R}^+.$$

Le coefficient de pondération k_f de la fonction économique éloigne, s'il est inférieur à 1, le centre du tronçon de la surface équipotentielle de la fonction, et le rejette donc vers l'optimum $\hat{x}$.
Il est possible d'ajuster, pour chaque problème particulier, le coefficient k_f de manière à accélérer le plus possible la convergence de la méthode.

D'autrepart, une des caractéristiques essentielles de la méthode des centres étant de trouver des points bien au centre des tronçons considérés, il est intéressant de donner une même importance à toutes les contraintes dans l'expression de la F-distance, de manière à obtenir une certaine symétrie.
Pour cela, les fonctions considérées sont normalisées, c'est-à-dire divisées par la norme de leur gradient au point de linéarisation **y**.

On remplace donc :

$$f(x) - \lambda \text{ par } \frac{f(x)-\lambda}{||\nabla f(y)||} \quad \text{et } g_i(x) \text{ par } \frac{g_i(x)}{||\nabla g_i(y)||}, \; i \in L.$$

On normalise de même les fonctions linéarisées ; on remplace donc

$$f'(x;y) - \lambda \text{ par } \frac{f'(x;y)-\lambda}{||\nabla f(y)||} \text{ et } g'_i(x;y) \text{ par } \frac{g'_i(x;y)}{||\nabla g_i(y)||}, \; i \in L.$$

Les contraintes additionnelles de centrage sont normalisées de la même manière, soit au point y, soit à leur point de linéarisation w.

On voit que la normalisation modifie les fonctions à chaque troncature. Les démonstrations de convergence de la méthode des centres doivent donc être légèrement modifiées pour tenir compte de cette normalisation. Les modifications nécessaires ont été réalisées dans [10] par l'introduction de fonctions majorantes de la F-distance.

I.6.2 *Obtention d'un point réalisable de départ.*

Nous reprenons ici l'algorithme fini exposé dans [23].
Considérons $A \subset \mathbb{R}^n$ et $B \subset \mathbb{R}^n$
tels que

$$\overset{\circ}{A} \cap B \neq \emptyset.$$

On se donne une famille E d'ensembles de $\mathbb{R}^n$ telle que $A \in E$ et d une F-distance régulière définie sur $\mathbb{R}^n \times E$.

On choisit une suite $\{\varepsilon_k \geq 0 \mid k \in \mathbb{N}\}$ tendant vers o quand $k \to \infty$.

Itération k :

On dispose de $E_k \in E$ et de $\overset{k}{x}$ tels que

$E_k \supset A$, $\overset{\circ}{E}_k \cap B \neq \emptyset$

$\overset{k}{x} \in \overset{\circ}{E}_k \cap B$.

$d(\overset{k}{x}, E_k) \geq \sup \{d(x, E_k) \mid x \in E_k \cap B\} - \varepsilon_k$.

- Si $\overset{k}{x} \in A \cap B$:
  prendre $E_{k+1} = E_k$ et $\overset{k+1}{x} = \overset{k}{x}$.

- Sinon :
  Choisir $E_{k+1} \in E$ tel que
  $E_k \supset E_{k+1} \supset A$, $\overset{k}{x} \notin E_{k+1} \cap B$

PROPOSITION I.8. : (algorithme fini).

$\exists k^* \in \mathbb{N}$ tel que
$k \geq k^* \Longrightarrow \overset{k}{\overset{\circ}{x}} \in \overset{\circ}{A} \cap B$.

Démonstration : [23].

PARTICULARISATION DE L'ALGORITHME.

Considérons le problème (P) défini précédemment.
L'ensemble A de l'algorithme fini sera le tronçon initial $E(\lambda_o)$, l'ensemble B le polyèdre linéaire, et d la F-distance du minimum des contraintes.

On se donne au départ un point $\overset{\circ}{x} \in B$, et on suppose que $\overset{\circ}{x} \notin \overset{\circ}{E}(\lambda_o)$

L'ensemble $E(\lambda_o)$ est défini par

$$E(\lambda_o) = \{x | f(x) - \lambda_o \geq o,\ g_i(x) \geq o, i \in L\}.$$

On se donne E en définissant les E_k par

$$E_k = \{x | f(x) - \lambda_o \geq \delta_k,\ g_i(x) \geq \delta_k,\ i \in L\}$$

avec

$$\delta_k \in \mathbb{R}^-.$$

Il est clair que $\delta_k > \delta_k' \Longrightarrow E_k \subset E_k'$.

D'après les définitions de d et E_k on a

$$d(x,E_k) = \min \{f(x) - \lambda_o - \delta_k,\ g_i(x) - \delta_k | i \in L\}$$

$$= \min \{f(x) - \lambda_o,\ g_i(x) | i \in L\} - \delta_k$$

$$= d(x,E(\lambda_o)) - \delta_k.$$

Les opérations à effectuer dans l'algorithme fini sont donc :

Déterminer $\overset{k}{x}$ tel que

$$d(\overset{k}{x}, E(\lambda_o)) - \delta_k \geq \sup \{d(x,E(\lambda_o)) - \delta_k \mid x \in E_k \cap B\} - \varepsilon_k$$

et si $\overset{k}{x} \notin \overset{\circ}{E}(\lambda_o) \cap B$, choisir E_{k+1} tel que

$$\delta_{k+1} = d(\overset{k}{x}, E(\lambda_o)).$$

ce qui peut s'écrire :

Déterminer $\overset{k}{x}$ tel que

$$d(\overset{k}{x}, E(\lambda_o)) \geq \sup \{d(x,E(\lambda_o)) \mid x \in E_k \cap B\} - \varepsilon_k.$$

Les points $\overset{k}{x}$ seront déterminés par linéarisation, comme dans l'algorithme partiel (I.4.2) :

Résoudre le programme mathématique

$$\begin{cases} \text{Maximiser } d'(x,\lambda_o;\overset{k-1}{x}) \\ x \in B \end{cases}$$

Soit $\overset{k-1}{z}$ une solution optimale . Déterminer $\overset{k}{x}$ tel que

$$d(\overset{k}{x},E(\lambda_o)) = \max \{d(x,E(\lambda_o)) \mid x \in [\overset{k-1}{x}, \overset{k-1}{z}]\}.$$

L'algorithme ainsi particularisé est le même que l'algorithme partiel de I.4.2. En effet on a vu que cet algorithme s'applique même dans le cas où le point de départ n'appartient pas au tronçon, la seule hypothèse nécessaire étant qu'il appartienne à B.

La proposition I.5. permet d'affirmer que la suite $\{\overset{k}{x}\}$ converge vers un centre du tronçon $E(\lambda_o) \cap B$, ce qui implique que $\{\varepsilon_k\} \to o$.

La proposition I.8. s'applique donc, ce qui démontre que l'algorithme particularisé est fini.

En pratique, ceci revient à effectuer plusieurs linéarisations successives du tronçon $E(\lambda_o)$; on obtiendra nécessairement un point réalisable au bout d'un nombre fini de linéarisations.

Remarque :

On peut, comme en I.6.1, pondérer la fonction économique par un coefficient $k_f < 1$.
L'expérience montre qu'en général le premier point réalisable est obtenu au bout d'un nombre plus grand de linéarisations, mais qu'il est plus proche de l'optimum $\hat{x}$ de (P).

CHAPITRE II

MAXIMISATION
DE LA F-DISTANCE LINEARISEE
SUR UN POLYEDRE LINEAIRE

II.1 Introduction.

Dans ce qui suit, nous utilisons l'algorithme de la proposition I.7., c'est-à-dire la méthode des centres linéarisée avec centrage.

Un des calculs essentiels de cet algorithme est la résolution du programme mathématique

$$Q'(\lambda;y) \quad \left\| \begin{array}{l} \text{Maximiser } d'(x,\lambda;y) \\ x \in B \end{array} \right.$$

avec y, le point de linéarisation, et $\lambda \in K$, fixés.
De même, à chaque étape de l'algorithme de centrage, on doit résoudre

$$QC'_{h+1}(\lambda;y) \quad \left\| \begin{array}{l} \text{Maximiser } d'_{h+1}(x,\lambda;y) \\ x \in B \end{array} \right.$$

avec $d'_{h+1}(x,\lambda;y) = \min\{d'_h(x,\lambda;y),\ l'(x;\overset{h}{w})\}$.
La maximisation, sur le polyèdre linéaire B, de la F-distance linéarisée, est donc un sous-problème qui se pose fréquemment, et qu'il convient d'optimiser du point de vue temps de calcul et encombrement mémoire.

Nous exposons dans ce chapitre deux approches possibles de la résolution de ce problème.

II.2 Utilisation de la méthode simpliciale.

II.2.1 *Généralités*.

De manière classique, le programme $Q'(\lambda;y)$ est équivalent au programme linéaire suivant, obtenu à l'aide d'une variable supplémentaire μ :

$$Q''(\lambda;y) \quad \left\| \begin{array}{l} \text{Maximiser } \mu \\ g'_i(x;y) - \mu \geq 0 \quad \forall i \in L \\ f'(x;y) - \mu \geq \lambda \\ x \in B \end{array} \right\} S_o$$

λ et y étant fixés.
On doit donc résoudre une suite de programmes linéaires $Q''(\lambda;y)$, de taille constante et de structure analogue à celle du problème d'origine.
De même, le programme $QC'_{h+1}(\lambda;y)$ est équivalent au programme linéaire.

$$QC''_{h+1}(\lambda;y) \quad \left\| \begin{array}{l} \text{Maximiser } \mu \\ (x,\mu) \in S_h \\ l'(x;\overset{h}{w})-\mu \geq 0 \\ x \in B \end{array} \right\} S_{h+1}$$

S_h étant le polyèdre en (x,μ) correspondant au programme $QC''_h(\lambda;y)$.

II.2.2 *Méthode simpliciale et duale-simpliciale en variables bornées.*

Le programme $Q''(\lambda;y)$ est de la forme

$$(P) \quad \left\| \begin{array}{l} \text{Maximiser } fx \\ Ax = a \\ x^m \leq x \leq x^M \end{array} \right.$$

où x^m et x^M sont les bornes respectivement inférieure et supérieure sur les variables.
La méthode simpliciale en variables bornées est donc bien adaptée à la résolution de ce problème.

Rappelons brièvement les caractéristiques de cette méthode.
Soient I une base, $\bar{I}$ l'ensemble des indices hors-base, et (B^+,B^-) une partition de $\bar{I}$. On considère la solution de base $x(I,B^+,B^-)$ définie par

$$\left| \begin{array}{l} x_{B^+} = x^M_{B^+} \\ x_{B^-} = x^m_{B^-} \\ x_I = t - T^{B^+} x^M_{B^+} - T^{B^-} x^m_{B^-} \end{array} \right.$$

avec $T = (A^I)^{-1}A$ et $t = (A^I)^{-1}a$.

Pour déterminer une variable candidate à entrer dans la base, on choisit $s \in B^-$: $d^s > o$, ou $s \in B^+$: $d^s < o$, d étant le critère de candidature $f - f^I T$.

1er cas : $s \in B^+$

On calcule $\theta = \min \{\alpha, \beta, \gamma\}$ avec

$$\alpha = \min \left\{ \frac{x_i^m - t_i + T_i^{B^+} x_{B^+}^M + T_i^{B^-} x_{B^-}^m}{T_i^s} \;\middle|\; i \in I,\ T_i^s < o \right\}$$

$$\beta = \min \left\{ \frac{x_i^M - t_i + T_i^{B^+} x_{B^+}^M + T_i^{B^-} x_{B^-}^m}{T_i^s} \;\middle|\; i \in I,\ T_i^s > o \right\}$$

$$\gamma = x_s^M - x_s^m$$

Si $\theta = \gamma$, la variable s passe de B^+ à B^- : il n'y a pas de changement de base.
Sinon, r est l'indice i donnant le rapport minimum ; on effectue le changement de base $I' = I + s - r$. Si $\theta = \alpha$, la variable r passe dans B^- ; si $\theta = \beta$ elle passe dans B^+.

2e cas : $s \in B^-$

On effectue des calculs analogues avec $-T_i^s$ au lieu de T_i^s.

Les programmes $QC''_{h+1}(\lambda;y)$ sont obtenus en ajoutant, à raison d'une à la fois, des contraintes au programme linéaire $Q''(\lambda;y)$. On connaît une solution optimale réalisable de $QC''_h(\lambda;y)$, qui est donc optimale non réalisable pour $QC''_{h+1}(\lambda;y)$, et on veut déterminer une solution optimale réalisable de $QC''_{h+1}(\lambda;y)$. La méthode duale-simpliciale est bien adaptée à la résolution d'un tel problème et permet une économie de calculs.
D'autre part, $QC''_{h+1}(\lambda;y)$ étant de la forme (P) précédente, il est logique d'utiliser la méthode duale-simpliciale en variables bornées.

Soient I une base, $\bar{I}$ l'ensemble des indices hors-base, et (B^+,B^-) une partition de $\bar{I}$. On considère la solution de base $x(I,B^+,B^-)$ définie par

$$\begin{cases} x_{B^+} = x^M_{B^+} \\ x_{B^-} = x^m_{B^-} \\ x_I = t - T^{B^+} x^M_{B^+} - T^{B^-} x^m_{B^-} \end{cases}$$

La variable qui sort de la base est $r \in I$: $x_r < x^m_r$, ou $r \in I$: $x_r > x^M_r$.

<u>1er cas</u> : $x_r < x^m_r$

La variable r passe de I à B^-.

On calcule $\theta = \min \{\alpha,\beta\}$ avec

$$\alpha = \min \left\{ \frac{d^j}{T^j_r} \,\middle|\, j \in B^+,\ T^j_r > o \right\}$$

$$\beta = \min \left\{ \frac{d^j}{T^j_r} \,\middle|\, j \in B^-,\ T^j_r < o \right\}$$

L'indice s de la variable candidate est l'indice j donnant le rapport minimum.

<u>2e cas</u> : $x_r > x^M_r$

La variable r passe de I à B^+.

On effectue des calculs analogues avec $-T^j_r$ au lieu de T^j_r.

II.2.3 *Méthode de l'inverse explicite* :

Nous utilisons, plutôt que la méthode du calcul complet du tableau simplicial à chaque itération, la méthode dite "révisée" ou "de l'inverse explicite" : on conserve en mémoire la matrice A et l'inverse de base $(A^I)^{-1}$, que l'on modifie à chaque changement de base.

Si le programme linéaire est de grande taille, la méthode de l'inverse explicite permet d'utiliser le creux de la matrice, qui devient de plus en plus important au fur et à mesure que la taille du problème augmente. On utilise pour cela des procédés de "compactification de l'inverse", qui consistent à ranger $(A^I)^{-1}$ en mémoire sous une forme plus condensée que la forme explicite, par factorisation en particulier, comme on le verra en II.2.4 et II.2.5.

Si l'on effectue le changement de base I' = I-r+s :

Soit $T^s = (A^I)^{-1}A^s$. On a les formules de transformation :

$$((A^{I'})^{-1})^j_i = ((A^I)^{-1})^j_i - \frac{T^s_i}{T^s_r}((A^I)^{-1})^j_r \qquad i,j \in I-r = I'-s$$

$$((A^{I'})^{-1})^j_s = \frac{1}{T^s_r}((A^I)^{-1})^j_r \qquad j \in I-r = I'-s$$

$$((A^{I'})^{-1})^s_i = ((A^I)^{-1})^r_i - \frac{T^s_i}{T^s_r}((A^I)^{-1})^r_r \qquad i \in I-r = I'-s$$

$$((A^{I'})^{-1})^s_s = \frac{1}{T^s_r}((A^I)^{-1})^r_r$$

Les éléments nécessaires au changement de base sont

$$d^{\bar{I}} = f^{\bar{I}} - f^I(A^I)^{-1}A^{\bar{I}}$$

$$x_I = (A^I)^{-1}(a - A^{B^+} x^M_{B^+} - A^{B^-} x^m_{B^-})$$

$$T^s = (A^I)^{-1} A^s.$$

II.2.4 *Factorisation de l'inverse.*

Soit J_m la matrice unité d'ordre m, e^k le $k^{\text{ème}}$ vecteur unitaire, et $(e^k)^t$ son transposé. On vérifie aisément les propriétés suivantes :

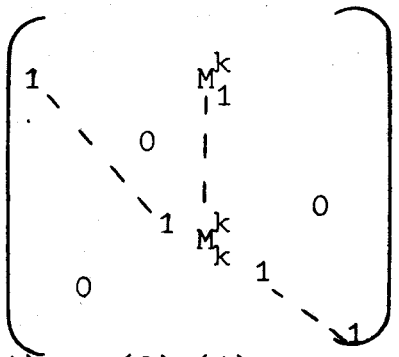
PROPOSITION II.1 :

Soit M = $\begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & & & \\ & & & 1 & M^k_1 & & \\ & & & & \vdots & & \\ & & & & M^k_k & & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & & & M^k_m & & \end{pmatrix}$ $= J_m + (M^k - e^k)(e^k)^t$

L'inverse de M est une matrice de la même forme :

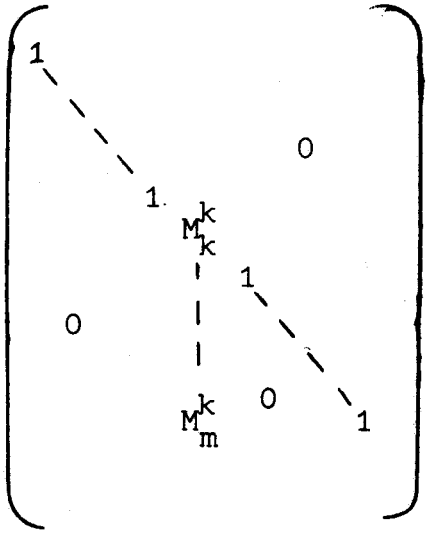
M^{-1} = $\begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & & & \\ & & & 1 & -\frac{M^k_1}{M^k_k} & & \\ & & & & \vdots & & \\ & & & & \frac{1}{M^k_k} & & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & & & -\frac{M^k_m}{M^k_k} & & \end{pmatrix}$ $= J_m - (M^k_k)^{-1}(M^k - e^k)(e^k)^t$

PROPOSITION II.2. :

Soit $M^{(k)}$ =  d'ordre m

Le produit $M^{(m)}M^{(m-1)}\ldots M^{(2)}M^{(1)}$ est une matrice triangulaire supérieure dont les éléments de la colonne k sont ceux de la matrice $M^{(k)}$.

PROPOSITION II.3. :

Soit $M^{(k)}$ =  d'ordre m

Le produit $M^{(1)}M^{(2)}\ldots M^{(m)}$ est une matrice triangulaire inférieure dont les éléments de la colonne k sont ceux de la matrice $M^{(k)}$.

La matrice de base A^I est indicée en lignes et en colonnes par I, avec |I| = m. Dans tout ce qui suit, on suppose que l'ensemble I est ordonné, c'est-à-dire qu'on définit un isomorphisme entre I et {1,...m}, tel qu'à tout k ∈ I il correspond un unique indice l ∈ {1,...m}.

FACTORISATION PAR DIAGONALISATION (P. F. I.)

Ce procédé est basé sur la méthode d'inversion de Jordan.
La matrice de base A^I étant d'ordre m, on peut écrire

$A^I = E^{(1)}E^{(2)}\ldots E^{(m)}$

d'où $(A^I)^{-1} = E^{(m)-1}\ldots E^{(2)-1}E^{(1)-1}$

chacune des matrices $E^{(I)}$, indicée en lignes et en colonnes par I, étant construite à partir d'une colonne de A^I, de la façon suivante :

$$E^{(I)} = \begin{pmatrix} 1 & & & \eta_1^k & & \\ & \ddots & & \vdots & & \\ & 0 & & \vdots & & \\ & & 1 & \eta_k^k & & 0 \\ & & & \vdots & \ddots & \\ 0 & & & \vdots & & 1 \\ & & & \eta_m^k & & \end{pmatrix} = J_m + (\eta^k - e^k)\,(e^k)^t$$

avec $\eta^k = E^{(1-1)-1}\ldots E^{(2)-1}E^{(1)-1}A^k$, $k \in I$.

D'où $E^{(1)-1} = J_m - (\eta_k^k)^{-1}\,(\eta^k - e^k)(e^k)^t$.

Les matrices $E^{(1)-1}$, étant des matrices unité à l'exception de leur colonne k, peuvent être définies en mémoire par l'indice k définissant la colonne non unitaire, ainsi que la valeur numérique des éléments non nuls dans la colonne et leur indice de ligne correspondant.

En pratique, pour obtenir une factorisation aussi condensée que possible, on échange les colonnes et les lignes de A^I de manière à rendre la matrice aussi proche que possible d'une matrice triangulaire inférieure :

$$A^I = \begin{pmatrix} L_1 & & \\ C & B & \\ D & E & L_2 \end{pmatrix}$$

avec L_1 et L_2 triangulaires inférieures et B aussi petite que possible

A^I peut alors être factorisée

$$A^I = \begin{pmatrix} L_1 & & \\ C & J_q & \\ D & & J_r \end{pmatrix} \begin{pmatrix} J_p & & \\ & B & \\ & & J_r \end{pmatrix} \begin{pmatrix} J_p & & \\ & J_q & \\ & E & J_r \end{pmatrix} \begin{pmatrix} J_p & & \\ & J_q & \\ & & L_2 \end{pmatrix}$$

L_1 et L_2 étant triangulaires inférieures, le calcul de la forme produit n'est non trivial que pour la sous-matrice B.

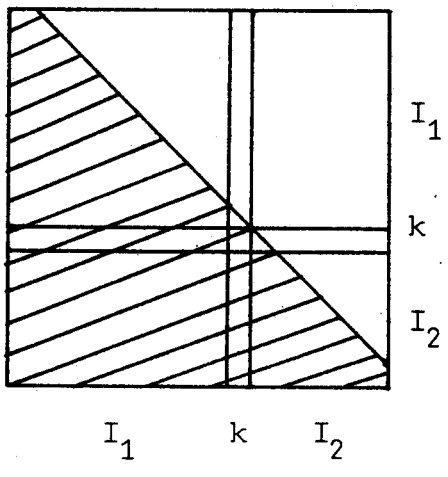
FACTORISATION PAR TRIANGULARISATION (E. F. I.)

Ce procédé est basé sur la méthode d'inversion de Gauss.
On sait qu'il existe une matrice L triangulaire inférieure et une matrice **U** triangulaire supérieure, indicées en lignes et en colonnes par I, telles que $A^I = LU$.

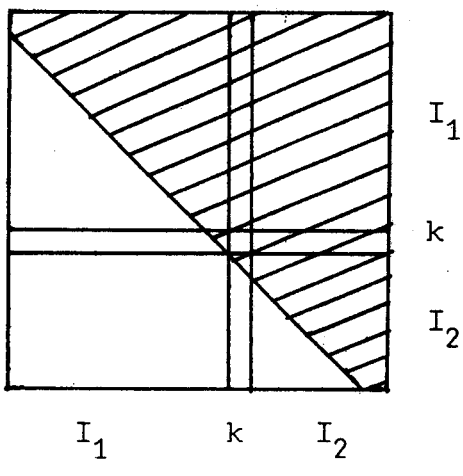
Soient L^k et U^k les colonnes d'indice k respectivement de L et **U**, $k \in I$.

Considérons une partition de I

L =



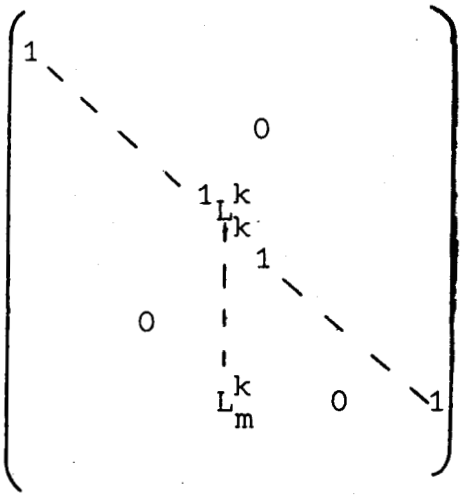
U =



$I = I_1 \cup \{k\} \cup I_2$

On peut factoriser L
$L = L^{(1)} L^{(2)} \ldots L^{(m)}$

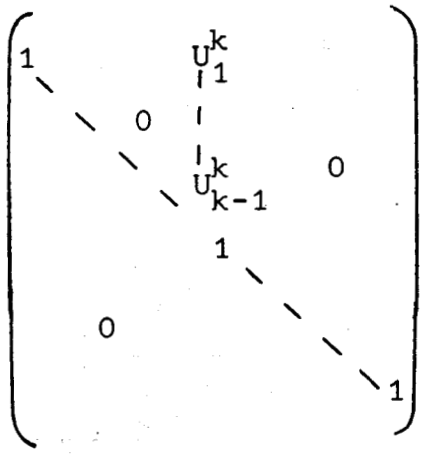
avec $L^{(l)} = J_m + (L^k - e^k)(e^k)^t =$



D'où $L^{(l)-1} = J_m - (L_k^k)^{-1}(L^k - e^k)(e^k)^t$.
Si {1,...l-1} est l'ensemble d'indices isomorphe à I1 :
Soit $\alpha^k = L^{(l-1)-1} \ldots L^{(2)-1} L^{(1)-1} A^k$
L^k et U^k sont données par
$L_i^k = o$, $i \in I_1$; $L_i^k = \alpha_i^k$, $i \in \{k\} \cup I_2$.
$U_i^k = \alpha_i^k$, $i \in I_1$; $U_k^k = 1$; $U_i^k = o$, $i \in I_2$

U peut également être factorisée en colonnes :
$U = U^{(m)} U^{(m-1)} \ldots U^{(2)} U^{(1)}$

avec $U^{(l)} = J_m + (U^k - e^k)(e^k)^t =$



D'où $U^{(l)-1} = J_m - (U^k - e^k)(e^k)^t$.

On a donc $(A^I)^{-1} = U^{(1)-1}....U^{(m)-1} \quad L^{(m)-1}...L^{(1)-1}$

Il est également possible de factoriser U en lignes :
$U = U^{(m)}U^{(m-1)}...U^{(1)}$

avec $U^{(1)} = J_m + e^k(U_k-(e^k)^t)$

$$= \begin{pmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & U_k^{k+1}....U_k^m & \\ & & & 0 & \\ 0 & & & & 1 \end{pmatrix}$$

D'où $U^{(1)-1} = J_m - e^k(U_k-(e^k)^t)$

Cette factorisation de U en lignes est intéressante dans le cas où A^I est symétrique ; en effet on a alors $U_i^j = \frac{L_i^j}{L_i^i}$, i, j $\in$ I, ce qui permet de ne conserver en mémoire que l'une des matrices L ou U, la factorisation de U étant symétrique de celle de L au coefficient L_i^i près. Si A^I n'est pas symétrique en valeur mais l'est en structure, la symétrie de structure est conservée, ce qui permet aussi un gain de place dans les procédés d'indiçage ([38]).

En pratique, pour obtenir une factorisation aussi condensée que possible, on échange les colonnes et les lignes de A^I de manière à obtenir la répartition :

$$A^I = \begin{pmatrix} L_1 & & \\ D' & L'_2 & E' \\ C & & B \end{pmatrix}$$

D' et E' étant les matrices D et E précédentes dont l'ordre des lignes a été inversé ; L'_2 étant la matrice L_2 précédente dont l'ordre des lignes et celui des colonnes ont été inversés.
L'_2 est donc triangulaire supérieure. Là encore, la seule factorisation non triviale est celle de B.

RELATION ENTRE P. F. I. ET E. F. I.

En utilisant les définitions précédentes, nous avons
$E^{(m)-1} \ldots E^{(1)-1} A^I = J_m$
$L^{(m)-1} \ldots L^{(1)-1} A^I = U$

Considérons la colonne non unitaire de $E^{(1)}$

$$\eta^k = \begin{pmatrix} \eta_1^k \\ \vdots \\ \eta_k^k \\ \vdots \\ \eta_m^k \end{pmatrix}$$

On définit les vecteurs V^k et W^k.
$V_i^k = o$, $i \in I_1$; $V_i^k = \eta_i^k$, $i \in \{k\} \cup I_2$.
$W_i^k = \eta_i^k$, $i \in I_1$; $W_i^k = o$, $i \in \{k\} \cup I_2$.

$$V^k = \begin{pmatrix} o \\ \vdots \\ o \\ \eta_k^k \\ \vdots \\ \eta_m^k \end{pmatrix} \qquad W^k = \begin{pmatrix} \eta_1^k \\ \vdots \\ \eta_{k-1}^k \\ o \\ \vdots \\ o \end{pmatrix}$$

et on définit les matrices
$V^{(1)} = J_m + (V^k - e^k)(e^k)^t$
$W^{(1)} = J_m + W^k (e^k)^t$.

Nous citons un résultat de [4] :

PROPOSITION II.4. :

On a $\eta^k = (L + J_m - U^{-1})^k$, $\forall k \in I$.

Démonstration :

a) On montre par récurrence que $V^{(l)} = L^{(l)}$ $\forall l = 1,...m$.
Il est évident que $L^{(1)} = E^{(1)} = V^{(1)}$.
Supposons que $V^{(j)} = L^{(j)}$ $\forall j$, $1 \leq j < l \leq m$.
On vérifie facilement que $V^{(j)}W^{(j)} = E^{(j)}$ $\forall j = 1,...m$
et que $V^{(i)}W^{(j)} = W^{(j)}V^{(i)}$ pour $1 \leq j < i \leq m$.

$$E^{(l-1)-1}...E^{(1)-1}A^I = W^{(l-1)-1}V^{(l-1)-1}...W^{(1)-1}V^{(1)-1}A^I.$$
$$= W^{(l-1)-1}...W^{(1)-1}V^{(l-1)-1}...V^{(1)-1}A^I$$
$$= W^{(l-1)-1}...W^{(1)-1}L^{(l-1)-1}...L^{(1)-1}A^I$$

$$\eta^k = E^{(l-1)-1}...E^{(2)-1}E^{(1)-1}A^k$$
$$= W^{(l-1)-1}...W^{(1)-1}L^{(l-1)-1}...L^{(1)-1}A^k$$
$$= W^{(l-1)-1}...W^{(1)-1}\alpha^k.$$

Les matrices $W^{(1)-1},...W^{(l-1)-1}$ sont de la forme $M^{(k)}$ de la proposition II.2., avec $M^k_k = 1$.
Donc $\eta^k_j = \alpha^k_j$ pour $j \in \{k\} \cup I_2$.
D'où d'après la définition de L^k, $\eta^k_j = L^k_j$ pour $j \in \{k\} \cup I_2$
Donc $L^{(l)} = V^{(l)}$

b) Comme précédemment on peut écrire

$$E^{(m)-1}...E^{(1)-1}A^I = J_m = W^{(m)-1}...W^{(1)-1}L^{(m)-1}...L^{(1)-1}A^I.$$
$$= W^{(m)-1}...W^{(1)-1}U.$$

Donc $U^{-1} = W^{(m)-1}...W^{(1)-1}$.
D'après la proposition II.2., les colonnes non unitaires des matrices $W^{(l)-1}$ sont égales aux colonnes de même indice de la matrice U^{-1}

D'autre part, $W^{(1)-1} = J_m - W^k (e^k)^t$; donc
$W^{(1)-1} = 2J_m - W^{(1)}$, d'où $(W^{(1)-1})^k = 2e^k - (W^{(1)})^k$.

On a donc

$$(L + J_m - U^{-1})^k = L^k + e^k - (U^{-1})^k$$
$$= (L^{(1)})^k + e^k - (W^{(1)-1})^k$$
$$= (L^{(1)})^k + (W^{(1)})^k - e^k$$

Or $L^{(1)} = V^{(1)} \Longrightarrow (L^{(1)})^k = (V^{(1)})^k = V^k$

D'autre part $(W^{(1)})^k = W^k + e^k$

Donc $(L + J_m + U^{-1})^k = V^k + W^k = n^k$.

L'encombrement mémoire de la méthode P F I est donc $L + J_m - U^{-1}$, alors que celui de la méthode E F I est $L - J_m + U$. Pour comparer la place occupée par ces deux méthodes il faut donc comparer le nombre d'éléments non nuls de U et U^{-1}.

Or il est montré dans [4] que, si les éléments diagonaux de A^I sont non nuls, la méthode E F I est minimale pour l'encombrement mémoire, c'est-à-dire qu'il n'existe pas de méthode calculant L et U^{-1} qui utilise moins de place (ceci implique en particulier que la forme factorisée de U déterminée par la méthode E F I n'a jamais plus d'éléments non nuls que la forme factorisée de U^{-1}). On voit donc l'intérêt de cette méthode.

Remarque :

Il existe d'autres méthodes de triangularisation, en particulier la méthode de Householder, qui utilise des matrices de transformation orthogonales, et la méthode de Givens, qui utilise des matrices de rotation. Ces méthodes semblent meilleures que l'élimination de Gauss du point de vue de la propagation des erreurs d'arrondi, mais elles ne sont pas minimales du point de vue de l'encombrement mémoire.

II.2.5 *Méthodes de compactification et de mise à jour de l'inverse de base.*

En nous inspirant de [35], nous citons quatre types de méthodes de rangement en mémoire de l'inverse de base. La première méthode utilise la factorisation, par diagonalisation et les trois autres la factorisation par triangularisation. A une étape k quelconque de la méthode simpliciale, la matrice A est indicée en lignes par I. A l'étape k + 1, après qu'on ait effectué le changement de base I' = I - r + s, la matrice A est indicée en lignes par I'. En fait, la matrice A n'a pas changé ; seule sa ligne r a été renumérotée s. Dans ces conditions on peut, par abus de langage, considérer indifféremment la matrice A comme indicée par I ou I' en identifiant les indices r et s. Dans ce qui suit, on utilisera cette convention partout où cela ne crée pas d'ambiguité.

1ère méthode :

On effectue le changement de base I' = I - r + s.
Soit $\eta^{l+1} = (A^I)^{-1} A^s$.
On a $(A^{I'})^{-1} = E^{(l+1)-1} (A^I)^{-1}$ avec $E^{(l+1)} = J_m + (\eta^{l+1} - e^r)(e^s)^t$.

La matrice $E^{(l+1)}$ est indicée en lignes par I et en colonnes par I' ; son inverse $E^{(l+1)-1}$ est donc indicée en lignes par I' et en colonnes par I.
D'où, si $(A^I)^{-1} = E^{(l)-1} \ldots E^{(2)-1} E^{(1)-1}$
$(A^{I'})^{-1} = E^{(l+1)-1} E^{(l)-1} \ldots E^{(1)-1}$.

La mise à jour de l'inverse de base consiste donc à ajouter une matrice $E^{(l)}$ à la fin de la liste.

Pour calculer une ligne ou une colonne du tableau simplicial, on effectue une suite de produits vecteur-matrice, de la gauche vers la droite ou de la droite vers la gauche, chaque produit ne modifiant au plus qu'un élément dans le vecteur ; de plus, les multiplications portant sur des éléments nuls ne sont pas effectuées.

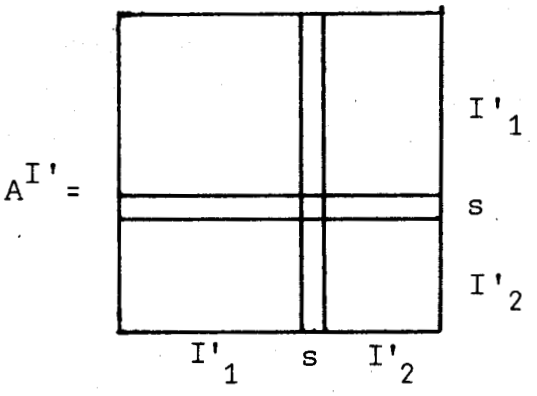
L'avantage principal de cette méthode est sa grande simplicité de mise en œuvre ; d'autre part elle conduit à un minimum du point de vue volume des calculs.

2e méthode.

On a $A^I = LU$, d'où $(A^I)^{-1} = U^{(1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1}$

On effectue le changement de base $I' = I - r + s$.

Considérons le partitionnement de I' :



avec $I' = I'_1 \cup \{s\} \cup I'_2$ (d'où $I = I'_1 \cup \{r\} \cup I'_2$)

$|I'_1| = k - 1$, $|I'_2| = m - k$.

Soient $U^{(1)}, \ldots U^{(k-1)}$ les facteurs correspondant aux colonnes d'indices $j \in I'_1$ de A^I, $U^{(k)}$ le facteur correspondant à la colonne r de A^I, et $U^{(k+1)}, \ldots U^{(m)}$ les facteurs correspondant aux colonnes d'indices $j \in I'_2$ de A^I.

Considérons la matrice $V = U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1} A^I$.

On a $V = U^{(k)} U^{(k-1)} \ldots U^{(1)}$.

La matrice $V' = U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1} A^{I''}$ est identique à V pour les colonnes $j \in I'_1 \cup I'_2$; sa colonne s est égale à

$V'^s = U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1} A^s$.

Soit $T^{(k)} = J_m + (V'^s - e^s)(e^s)^t$.

En factorisant la matrice V' on obtient
$V' = T^{(k)} U^{(k-1)} \ldots U^{(1)}$
D'où $(A^{I'})^{-1} = U^{(1)-1} \ldots U^{(k-1)-1} T^{(k)-1} U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1}$.

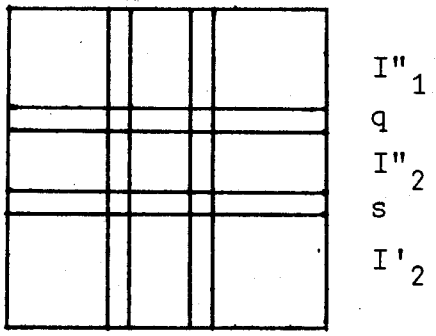
La factorisation de $(A^{I'})^{-1}$ est donc obtenue à partir de celle de $(A^{I})^{-1}$ en remplaçant le facteur $U^{(k)-1}$ par $T^{(k)-1}$.

Lors du changement de base suivant, la mise à jour de l'inverse peut s'effectuer différemment. Soit I" = I' - p + q ce changement de base. 2 cas sont à considérer :

1er cas : $p \in I'_1 \cup \{s\}$.

Considérons le partitionnement de I".

$A^{I"} =$



avec $I" = I"_1 \cup \{q\} \cup I"_2 \cup \{s\} \cup I'_2$ (d'où $I' = I"_1 \cup \{p\} \cup I"_2 \cup \{s\} \cup I'_2$)

$|I"_1| = 1 - 1$ (donc $1 \leq k$).

Soient $U^{(1)}, \ldots U^{(l-1)}$ les facteurs correspondant aux colonnes d'indices $j \in I"_1$ de $A^{I'}$, $U^{(l)}$ le facteur correspondant à la colonne p de $A^{I'}$, et $U^{(l+1)}, \ldots U^{(k-1)}$ les facteurs correspondant aux colonnes d'indices $j \in I"_2$ de $A^{I'}$.

On peut écrire
$U^{(l)} U^{(l-1)} \ldots U^{(1)} = U^{(l+1)-1} \ldots U^{(k-1)-1} T^{(k)-1} U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots$
$\ldots L^{(1)-1} A^{I'}$

D'où en raisonnant comme précédemment
$U^{(l+1)-1} \ldots U^{(k-1)-1} T^{(k)-1} U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1} A^{I"} =$
$T^{(l)} U^{(l-1)} \ldots U^{(1)}$.

avec $T^{(l)} = J_m + (V''^q - e^q)(e^q)^t$

et $V''^q = U^{(l+1)-1} \ldots U^{(k-1)-1} T^{(k)-1} U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1} A^q$.

D'où $(A^{I''})^{-1} = U^{(1)-1} \ldots U^{(l-1)-1} T^{(l)-1} U^{(l+1)-1} \ldots U^{(k-1)-1} T^{(k)-1}$
$U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1}$.

Dans ce cas la factorisation de $(A^{I''})^{-1}$ est donc obtenue à partir de celle de $(A^{I'})^{-1}$ en remplaçant le facteur $U^{(l)-1}$ par $T^{(l)-1}$.

2e cas : $p \in I'_2$.

On a $U^{(k-1)} \ldots U^{(1)} = T^{(k)-1} V'$

La matrice $V'' = T^{(k)-1} U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1} A^{I''}$ est identique à $T^{(k)-1} V'$ pour les colonnes d'indices $j \in I' - p = I'' - q$: sa colonne q est égale à $V''^q = T^{(k)-1} U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots L^{(1)-1} A^q$.

Soit $T^{(l)} = J_m + (V''^q - e^q)(e^q)^t$

En factorisant la matrice V'' on obtient

$V'' = T^{(l)} U^{(k-1)} \ldots U^{(1)}$

D'où $(A^{I''})^{-1} = U^{(1)-1} \ldots U^{(k-1)-1} T^{(l)-1} T^{(k)-1} U^{(k+1)-1} \ldots U^{(m)-1} L^{(m)-1} \ldots$
$\ldots L^{(1)-1}$.

Dans ce cas la factorisation de $(A^{I''})^{-1}$ est donc obtenue à partir de celle de $(A^{I'})^{-1}$ en ajoutant à gauche de $T^{(k)-1}$ le facteur $T^{(l)-1}$, et en ne supprimant aucun facteur.

On voit que cette méthode conduit à des problèmes de mise à jour du fichier U^{-1} car il faut intercaler des facteurs $T^{(k)-1}$ dans la factorisation, ce qui oblige à chaque fois à recopier le fichier ou à utiliser des techniques d'indiçage complexes et coûteuses.

D'autre part cette méthode, bien que basée au départ sur une décomposition triangulaire, utilise une technique analogue à la méthode 1 et lui est comparable du point de vue croissance du nombre d'éléments non nuls. Elle semble donc ne présenter qu'un intérêt limité.

3e méthode :

On a $A^I = LU$, d'où $(A^I)^{-1} = U^{-1}L^{(m)-1}\ldots L^{(1)-1}$.

On effectue le changement de base $I' = I - r + s$

Considérons le partitionnement de I' comme dans la méthode 2 :

$I' = I'_1 \cup \{s\} \cup I'_2$ (d'où $I = I'_1 \cup \{r\} \cup I'_2$)

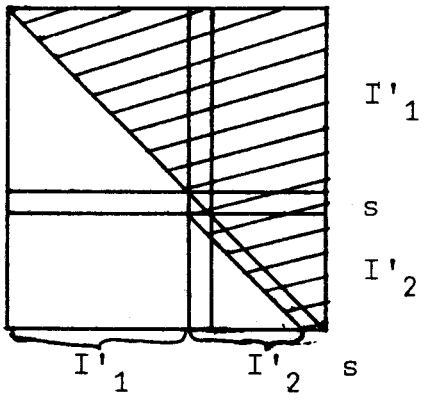
$|I'_1| = k - 1$, $|I'_2| = m - k$

La matrice $L^{-1}A^{I'}$ est identique à la matrice $U = L^{-1}A^I$ pour les colonnes $j \in I'_1 \cup I'_2$; sa colonne s est égale à $L^{-1}A^s$.

Dans la matrice $L^{-1}A^{I'}$, on réordonne les indices de colonnes de manière à amener la colonne s en position m. On obtient ainsi une matrice de Hessenberg.

$H = L^{-1}A^{I'}Q$ Q matrice de permutation

H =

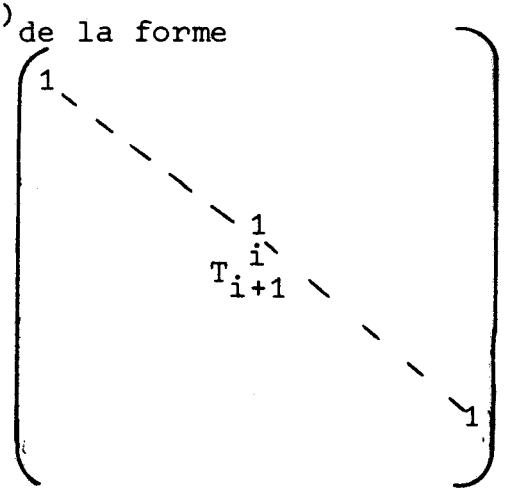


Les éléments sous la diagonale dans les colonnes d'indices $j \in I'_2$ sont alors éliminés par un ensemble de transformations élémentaires ; on obtient une nouvelle matrice triangulaire supérieure

$U' = V^{(m-1)}\ldots V^{(k)}H$

avec les matrices $V^{(l)}$ de la forme

$T^{(i)} =$



Les matrices $V^{(l)}$ sont en nombre m - k. Elles comportent en-dehors de la diagonale un seul élément non nul et il suffit de ranger en mémoire la valeur de cet élément et son indice de colonne.

On obtient finalement
$(A^{I'})^{-1} = Q\mathbf{U}'^{-1}V^{(m-1)}\ldots V^{(k)}L^{-1}$.
ce qui donne une nouvelle factorisation de l'inverse de base.

Il suffit d'ajouter les matrices $V^{(l)}$ à gauche du fichier L^{-1}.
Il est par contre plus difficile de mettre à jour le fichier $\mathbf{U}^{-1}$ écrit sous forme condensée, car le nombre d'éléments non nuls de $\mathbf{U}^{-1}$ croît.

D'autre part, plus le nombre d'éléments non nuls de $\mathbf{U}^{-1}$ croît et plus le volume des calculs par itération augmente.
Cependant, cette méthode, étant basée sur la triangularisation, est, du point de vue croissance du nombre d'éléments non nuls, meilleure que les méthodes 1 et 2.

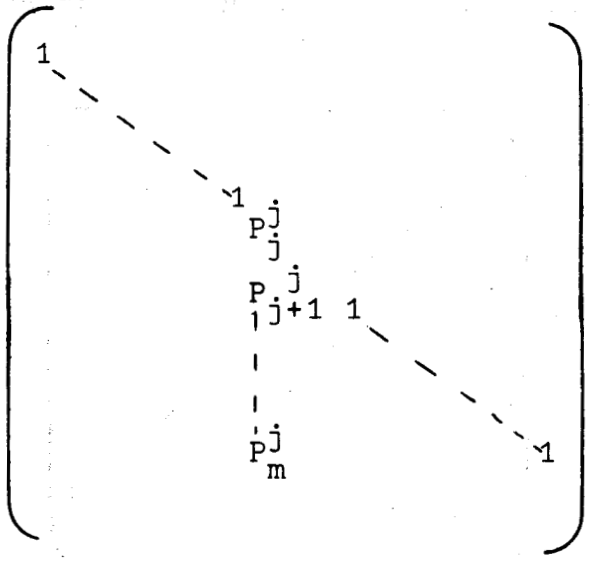
Remarque 1 :

avant chaque transformation $V^{(l)}$ on peut, par permutation de lignes, amener sur la diagonale l'élément dont la valeur est la plus grande possible, pour minimiser les erreurs d'arrondi. Il faut alors utiliser des matrices de permutation.

Remarque 2 :

les matrices $L^{(l)-1}$ sont de la forme

$P^{(j)} =$



On montre facilement les 3 propriétés suivantes, analogues aux propriétés citées dans [9]

(1) $T^{(i)}P^{(j)} = P^{(j)}T^{(i)}$ si $j>i+1$

Par commutations successives on peut donc amener la matrice $T^{(i)}$ à gauche de $P^{(i+1)}$

(2) $T^{(i)}P^{(i+1)} = P^{(i+1)}P'^{(i)}$

avec $P'^{(i)}$ matrice de la forme $P^{(i)}$, définie par :

$$P'^{i}_{i} = 1,\ P'^{i}_{i+1} = \frac{T^{i}_{i+1}}{P^{i+1}_{i+1}}\ ;\ P'^{i}_{j} = -\frac{P^{i+1}_{j}}{P^{i+1}_{i+1}}\,T^{i}_{i+1} \quad \forall j > i+1$$

On peut donc remplacer le produit $T^{(i)}P^{(i+1)}P^{(i)}$ par $P^{(i+1)}P'^{(i)}P^{(i)}$.

(3) $P'^{(i)}P^{(i)} = P''^{(i)}$

avec $P''^{(i)}$ matrice de la forme $P^{(i)}$, définie par :

$P''^{i}_{i} = P'^{i}_{i}P^{i}_{i}$; $P''^{i}_{j} = P'^{i}_{j}P^{i}_{i} + P^{i}_{j}$ $\forall j>i$.

On voit donc qu'en utilisant ces propriétés on peut calculer une nouvelle matrice L'^{-1} telle que $L'^{-1} = V^{(m-1)}\ldots V^{(k)}L^{-1}$ et d'avoir en mémoire la factorisation de cette matrice.
Ceci permet de conserver une décomposition triangulaire. Toutefois ce n'est pas essentiel à la méthode. On peut remarquer que cela conduit à une augmentation du nombre d'éléments non nuls de L^{-1} ; il faudra donc modifier le fichier L^{-1}, d'où les mêmes inconvénients que ceux signalés plus haut pour le fichier U^{-1}.

4e méthode :

On a $A^{I} = LU$, d'où $(A^{I})^{-1} = U^{(1)-1}\ldots U^{(m)-1}L^{(m)-1}\ldots L^{(1)-1}$.

On effectue le changement de base $I' = I - r + s$.

Considérons le partitionnement de I' comme dans les méthodes 2 et3.

$I' = I'_1 \cup \{s\} \cup I'_2$ (d'où $I = I'_1 \cup \{r\} \cup I'_2$)

$|I'_1| = k - 1$, $|I'_2| = m - k$

Comme dans la méthode 3, la matrice $L^{-1}A^{I'}$ est identique à la matrice $U = L^{-1}A^{I}$ pour les colonnes $j \in I'_1 \cup I'_2$; sa colonne s est égale à $L^{-1}A^{s}$. On va chercher à rendre $L^{-1}A^{I'}$ triangulaire supérieure.

On commence par annuler les éléments de la ligne s de la matrice dans les colonnes $j \in I'_2$. Pour cela on forme le vecteur de dimension m
$(v^k)^t = (0, U_s^{I'_2})U^{-1}$.

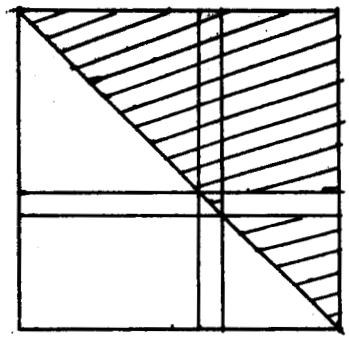
On considère la matrice $V^{(k)} = J_m + e^s(v^k)^t$, d'où
$V^{(k)-1} = J_m - e^s(v^k)^t$, et on forme la matrice $V^{(k)-1}L^{-1}A^{I'}$.

Les colonnes d'indices $j \in I'_1 \cup \{s\}$ de $L^{-1}A^{I'}$ sont inchangées par la transformation $V^{(k)-1}$, car les éléments de $(v^k)^t$ sont nuls dans les colonnes $j \in I'_1 \cup \{s\}$. Considérons les colonnes d'indices $j \in I'_2$: en ces colonnes $L^{-1}A^{I'}$ est identique à U, donc $V^{(k)-1}L^{-1}A^{I'}$ est identique à $V^{(k)-1}U$.

Or on a
$$V^{(k)-1}U = (J_m - e^s(v^k)^t)\,U = U - e^s(v^k)^tU$$
$$= U - e^s(0, U_s^{I'_2})$$

La matrice $V^{(k)-1}L^{-1}A^{I'}$ est donc identique à $L^{-1}A^{I'}$ sauf en sa ligne s dont les éléments appartenant aux colonnes $j \in I'_2$ ont été annulés.

$V^{(k)-1}L^{-1}A^{I'} =$



Soit alors
$T^s = V^{(k)-1}L^{-1}A^{s}$ et $T^{(k)} = J_m + (T^s - e^s)(e^s)^t$.

Si on applique la transformation $T^{(k)-1}$ à $V^{(k)-1}L^{-1}A^{I'}$, la colonne s de la matrice devient unitaire, et les colonnes d'indices $j \in I'_2$ sont inchangées puisqu'elles ont un 0 en ligne s. On obtient donc une matrice triangulaire supérieure U'.

$U' = T^{(k)-1}V^{(k)-1}L^{-1}A^{I'}$.

U' est obtenue à partir de U en remplaçant la ligne s et la colonne r de U par le vecteur unité $(e^r)^t$ ou e^s.

D'où $(A^{I'})^{-1} = U'^{-1}T^{(k)-1}V^{(k)-1}L^{-1}$.

U'^{-1} se déduit de U^{-1} comme U' de U et sa factorisation est immédiate.

Il semble qu'en général le nombre d'éléments non nuls créés dans les matrices $T^{(k)}$ et $V^{(k)}$ soit comparable au nombre des nouveaux éléments créés dans la méthode 3.

Mais ici le nombre d'éléments de U^{-1} décroît à chaque itération, ce qui montre que la méthode 4 est supérieure du point de vue encombrement mémoire. Il faut toutefois remarquer qu'il peut être difficile de "récupérer" la place laissée libre dans U^{-1} par la création de zéros. La mise à jour du fichier L^{-1} ne présente pas de difficultés car les facteurs $T^{(k)-1}V^{(k)-1}$ se placent à gauche de la factorisation de L^{-1}. Mais le volume de calculs nécessaire pour calculer $(v^k)^t$ croît à chaque itération et peut devenir important.

Remarque :

avant d'annuler les éléments des colonnes $j \in I'_2$ de la ligne s de la matrice $L^{-1}A^{I'}$, on peut, comme dans la méthode 3, transformer $L^{-1}A^{I'}$ en matrice de Hessenberg en amenant la colonne s en position m :

$H = L^{-1}A^{I'}Q$.

Puis on annule les éléments des colonnes $j \in I'_2$ de la ligne s de H en formant $V^{(k)-1}H = V^{(k)-1}L^{-1}A^{I'}Q$.

Pour obtenir une matrice triangulaire supérieure U', il suffit alors d'amener la ligne s en position m.

D'où $(A^{I'})^{-1} = QU'^{-1}Q^{-1}V^{(k)-1}L^{-1}$.

Soit $U^{(k)}$ le facteur correspondant à la colonne r de U.
Pour obtenir $U'^{-1}Q^{-1}$ à partir de la factorisation de U^{-1}, il suffit de supprimer de cette factorisation le facteur $U^{(k)-1}$, d'annuler tous les éléments correspondant à la ligne r de U^{-1}, et d'ajouter, à droite de la factorisation de U^{-1}, un nouveau facteur $U^{(m)-1}$ construit à partir du vecteur $T^s = V^{(k)-1}L^{-1}A^s$.

On voit que dans cette variante la matrice $T^{(k)-1}$, c'est-à-dire le nouveau facteur $U^{(m)-1}$, s'intègre dans la factorisation de U^{-1}. Ceci a un double avantage : d'une part le volume des calculs nécessaires pour calculer $(v^k)^t$ est moins important ; d'autre part le calcul de T^s n'utilise plus les matrices $T^{(k)}$ des itérations précédentes, ce qui réduit la croissance du nombre d'éléments non nuls. Les expériences numériques rapportées dans [14] montrent que cette méthode donne d'excellents résultats.

II.2.6 *Eliminations ordonnées*.

Dans la méthode simpliciale, le pivot est déterminé par l'indice de la variable qui entre dans la base et celui de la variable qui en sort. Le choix du pivot n'est donc pas libre.

Mais, de manière classique, pour minimiser l'accumulation des erreurs d'arrondi et pour diminuer la croissance du nombre d'éléments non nuls, on réinverse périodiquement la matrice de base A^I.

On peut même, au lieu de mettre à jour A^I à chaque changement de base, la réinverser à chaque itération. On a alors une méthode simpliciale "avec calculs directs" (nous détaillerons ce procédé en II.2.7.)

Dans ce cas, le choix des pivots successifs est libre. Un exemple simple montre qu'on peut les choisir de manière à minimiser la croissance du nombre d'éléments non nuls lors de l'inversion :

Soit la matrice

$$M = \begin{bmatrix} * & * & * & * \\ * & * & & \\ * & & * & \\ * & & & * \end{bmatrix}$$

où * représente un élément non nul.

Triangularisons cette matrice par élimination de Gauss.

Si nous éliminons les variables dans l'ordre 1,2,3, les pivots étant choisis sur la diagonale, la matrice aura les structures successives suivantes :

$$M_1 = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \end{bmatrix} \quad M_2 = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & * & * \end{bmatrix} \quad M_3 = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & & * \end{bmatrix}$$

On voit que la matrice triangulaire supérieure obtenue est entièrement remplie de termes non nuls.

Si par contre nous éliminons les variables dans l'ordre 4,3,2, les pivots étant choisis sur la diagonale, la matrice aura les structures successives suivantes :

$$M = \begin{bmatrix} * & & & * \\ & * & & * \\ & & * & * \\ * & * & * & * \end{bmatrix} \quad M_1 = \begin{bmatrix} * & & & * \\ & * & & * \\ & & * & * \\ & * & * & * \end{bmatrix} \quad M_2 = \begin{bmatrix} * & & & * \\ & * & & * \\ & & * & * \\ & & * & * \end{bmatrix} \quad M_3 = \begin{bmatrix} * & & & * \\ & * & & * \\ & & * & * \\ & & & * \end{bmatrix}$$

La matrice triangulaire supérieure obtenue est beaucoup plus creuse. D'autre part, les matrices intermédiaires étant également plus creuses, les matrices de transformation le sont aussi. L'encombrement-mémoire de la 2e décomposition est donc plus faible. L'ordre optimal de choix des pivots dépend de la structure de la matrice considérée. En pratique on utilise diverses techniques, dont aucune n'est optimale mais qui améliorent largement les performances dans ce domaine.

1er cas :

Les pivots doivent être choisis sur la diagonale.
C'est en particulier le cas si A^I est symétrique, en valeur ou en structure.

1e méthode :

on numérote les lignes de A^I en fonction du nombre de termes non nuls qu'elles contiennent, la ligne n°1 étant la plus creuse. L'ordre des pivots est I'ordre de cette numérotation.

2e méthode :

à chaque étape k de l'élimination on choisit comme ligne du pivot la ligne la plus creuse de la matrice $A^{(k)}$ en cours de transformation.

Cette méthode est très simple et donne de très bons résultats.
Elle est fréquemment employée (par exemple [38]).

3e méthode ([7]) : à chaque étape k on choisit le pivot tel que la transformation effectuée crée le plus petit nombre d'éléments non nuls :

Soit $c_i^{(k)}$ le nombre d'éléments non nuls de la ligne i de $A^{(k)}$ (non compris l'élément diagonal), $d_i^{(k)}$ le nombre d'éléments non nuls de la colonne i de $A^{(k)}$ (non compris l'élément diagonal), $M_i^{(k)}$ la sous-matrice formée des intersections des $d_i^{(k)}$ lignes correspondant aux termes non nuls de la colonne i avec les $c_i^{(k)}$ colonnes correspondant aux termes non nuls de la ligne i (non comprises la ligne et la colonne i), et $\alpha_i^{(k)}$ le nombre d'éléments non nuls de $M_i^{(k)}$. Le nombre d'éléments créés par pivotage sur $A_{ii}^{(k)}$ est $P_i^{(k)} = c_i^{(k)} d_i^{(k)} - \alpha_i^{(k)} - c_i^{(k)} - d_i^{(k)} - 1$.
On choisit i tel que $P_i^{(k)}$ soit minimum.

2e cas : cas général ([33])

Soit $c_i^{(k)}$ le nombre d'éléments non nuls en ligne i de $A^{(k)}$.
Pour toutes les colonnes j de $A^{(k)}$, on calcule $D_j^{(k)} = \Sigma c_i^{(k)}$, la somme étant faite sur les lignes correspondant aux éléments non nuls dans la colonne j de $A^{(k)}$, et on choisit j tel que $D_j^{(k)}$ soit minimum.
Pour déterminer la ligne i du pivot, on choisit le i qui minimise $c_i^{(k)}$.

II.2.7. *Méthode simpliciale avec calcul directs.*

Comme nous l'avons remarqué précédemment, tous les programmes linéaires obtenus par linéarisation possédent une structure comparable, et on remarque que, d'une linéarisation à l'autre, et même d'une troncature à l'autre, la base optimale de ces programmes linéaires change peu, et devient même en général constante au bout de quelques itérations. Si nous considérons, par exemple, la résolution du problème n°4 de l'annexe A1, nous obtenons la suite de bases optimales suivante

Troncature	Base optimale
1	3,6,10,12,14
2	2,3,7,10,14
3	2,3,7,10,14

La base optimale est ensuite constante jusqu'à l'optimum.

Il est donc intéressant de pouvoir "forcer" cette base, c'est-à-dire de partir, pour chaque programme linéaire, de la base optimale du programme linéaire précédent, et on peut penser qu'on arrivera ainsi au véritable optimum, soit directement, soit au bout de très peu de changements de base, ce qui permettra un gain de temps de calcul appréciable.
D'autre part, nous avons vu, dans le paragraphe précédent, que pour minimiser la croissance du nombre d'éléments non nuls à enregistrer en mémoire, on a intérêt à réinverser la matrice de base A^I le plus souvent possible.
Ces deux considérations nous ont conduit à adopter une méthode simpliciale avec calculs directs. Les éléments nécessaires aux changements de base sont calculés comme solutions de systèmes d'équations linéaires.

Le programme linéaire étant donné par

$$(P) \quad \left\| \begin{array}{l} \text{Maximiser } fx \\ Ax = a \\ x^m \leqslant x \leqslant x^M \end{array} \right.$$

la solution de base est $x_I = (A^I)^{-1}(a - A^{B^+} x^M_{B^+} - A^{B^-} x^m_{B^-})$

d'où $A^I x_I = a - A^{B^+} x^M_{B^+} - A^{B^-} x^m_{B^-}$

la colonne candidate est $T^s = (A^I)^{-1} A^s$

d'où $A^I T^s = A^s$

le critère de candidature est $d^{\bar{I}} = f^{\bar{I}} - uA^{\bar{I}}$ avec $u = f^I(A^I)^{-1}$

d'où $(A^I)^t u^t = (f^I)^t$.

On doit donc résoudre pour une base donnée deux systèmes avec la matrice A^I et un système avec sa transposée.

Comme nous l'avons signalé, nous voulons résoudre le programme (P) en partant d'une base I imposée quelconque.

Si la solution de base correspondante est réalisable, il suffit d'effectuer des changements de base en appliquant la méthode simpliciale, jusqu'à l'optimum.

Si la solution de base correspondante n'est pas réalisable, on ne peut appliquer directement la méthode simpliciale ; comme cette solution n'est pas non plus (en général) optimale, on ne peut pas davantage utiliser la méthode duale-simpliciale. Dans ce cas, nous devrons donc utiliser un procédé différent.

RESOLUTION EN DEUX PHASES.

La solution, étant de base, vérifie Ax = a. Par contre, n'étant pas réalisable, elle ne vérifie pas $x^m \leqslant x \leqslant x^M$. Dans une première phase, nous allons chercher une solution réalisable.

Pour cela, considérons la fonction économique annexe

$$\phi_1(x) = \sum_{j \in J_1} x_j - \sum_{k \in K_1} x_k$$

avec J_1 ensemble des j tels que $x_j < x_j^m$

K_1 ensemble des k tels que $x_k > x_k^M$

et posons de nouvelles bornes

$$\overset{1}{x}{}^{m}_{j} = x^{m}_{j} \quad \forall j \notin J_1, \quad \overset{1}{x}{}^{m}_{j} < x_j \quad \forall j \in J_1$$

$$\overset{1}{x}{}^{M}_{k} = x^{M}_{k} \quad \forall k \notin K_1, \quad \overset{1}{x}{}^{M}_{k} > x_k \quad \forall k \in K_1$$

Considérons le programme linéaire annexe

$$(P_1) \qquad \left\| \begin{array}{l} \text{Maximiser } \phi_1(x) \\ Ax = a \\ \overset{1}{x}{}^{m} \leq x \leq \overset{1}{x}{}^{M} \end{array} \right.$$

La solution x précédente est une solution de base réalisable pour (P_1). A partir de cette solution nous pouvons donc appliquer la méthode simpliciale et trouver l'optimum $\hat{x}$ de (P_1).

En $\hat{x}$ nous aurons nécessairement

$$\hat{x}_j = \overset{1}{x}{}^{M}_{j} \quad \forall j \in J_1$$

$$\hat{x}_k = \overset{1}{x}{}^{m}_{k} \quad \forall j \in K_1$$

$$\overset{1}{x}{}^{m}_{i} \leq \hat{x}_i \leq \overset{1}{x}{}^{M}_{i} \quad \forall i \notin J_1 \cup K_1$$

Comme d'autre part nous avons

$\forall j \in J_1, \; x_j < x^{m}_{j} < x^{M}_{j} \Longrightarrow \overset{1}{x}{}^{M}_{j} = x^{M}_{j}$

$\forall k \in K_1, \; x_k > x^{M}_{k} > x^{m}_{k} \Longrightarrow \overset{1}{x}{}^{m}_{k} = x^{m}_{k}$

$\forall i \notin J_1 \cup K_1, \; x^{m}_{i} \leq x_i \leq x_i{}^{M} \Longrightarrow \overset{1}{x}{}^{m}_{i} = x^{m}_{i}$ et $\overset{1}{x}{}^{M}_{i} = x^{M}_{i}$.

nous en déduisons

$x^{m} \leq \hat{x} \leq x^{M}$.

Donc $\hat{x}$ est une solution de base réalisable de (P).

Dans la pratique, il est probable qu'il n'est pas nécessaire d'aller jusqu'à l'optimum du problème (P_1), et qu'au bout de seulement quelques itérations de la résolution du programme annexe on aura une solution réalisable de (P).
On procédera donc de la façon suivante :
On effectue autant d'itérations qu'il est nécessaire pour que l'une des variables x_j ou x_k devienne réalisable pour (P).
Au point x obtenu on définit de nouveaux ensembles

$$J_2 = \{j | x_j < x_j^m\}, \quad K_2 = \{k | x_k > x_k^M\}$$

et une nouvelle fonction économique annexe

$$\phi_2(x) = \sum_{j \in J_2} x_j - \sum_{k \in K_2} x_k .$$

On pose de nouvelles bornes

$$x_j^{2m} = x_j^m \quad \forall k \notin J_2, \quad x_j^{2m} < x_j \quad \forall j \in J_2$$

$$x_k^{2M} = x_k^M \quad \forall k \notin K_2, \quad x_k^{2M} > x_k \quad \forall k \in K_2$$

et on considére le nouveau programme linéaire annexe

$$(P_2) \quad \left\| \begin{array}{l} \text{Maximiser } \phi_2(x) \\ Ax = a \\ x^{2m} \leq x \leq x^{2M} \end{array} \right.$$

Et ainsi de suite. Au bout d'un nombre fini d'étapes, on obtiendra des ensembles J_h et K_h vides, c'est-à-dire que toutes les variables seront comprises entre leurs bornes.
Dans une deuxième phase, à partir de la solution de base réalisable de (P) ainsi obtenue, on applique la méthode simpliciale jusqu'à l'optimum.

Remarque :

les variables x_j telles que $x_j < x_j^m$ deviennent, à l'optimum de (P_1), égales à leur borne supérieure. Il est donc possible de leur imposer de ne jamais passer à leur borne inférieure. De même, il est possible d'imposer aux variables x_k telles que $x_k > x_k^M$ de ne jamais passer à leur borne supérieure. On voit donc que dans ces conditions, il est inutile, en pratique, de modifier la valeur des bornes non vérifiées.

RESOLUTION EN UNE PHASE MIXTE.

Pour minimiser le volume des calculs, il est intéressant d'effectuer en même temps les deux phases précédentes.
Considérons pour cela le programme linéaire.

$$(P'_h) \qquad \begin{cases} \text{Maximiser } fx + \delta\phi_h(x) \\ Ax = a \\ x^{hm} \leq x \leq x^{hM} \end{cases}$$

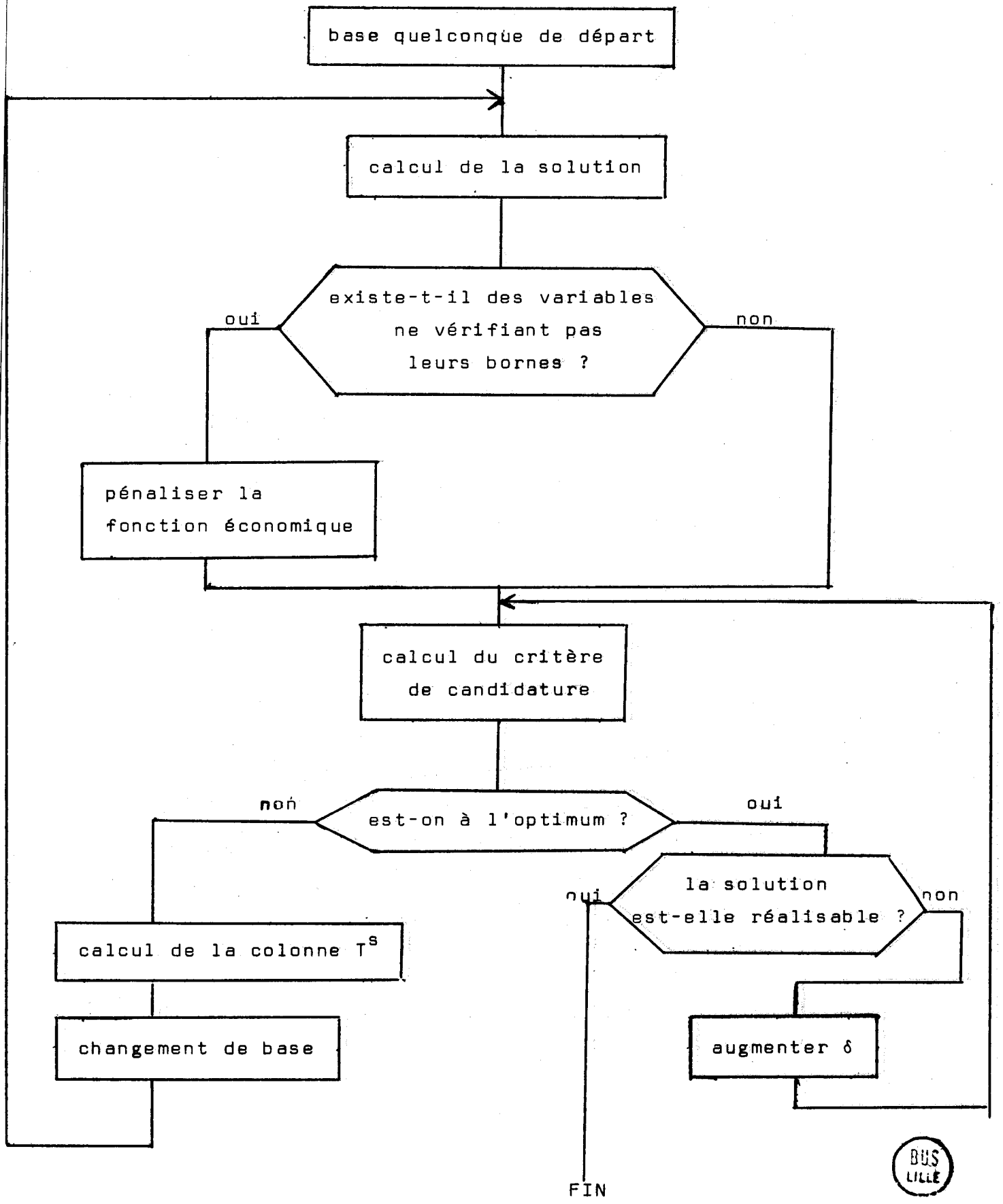
où $\delta \in \mathbb{R}^+$.

Comme précédemment, on effectue autant d'itérations de la méthode simpliciale qu'il est nécessaire pour que l'une des variables x_j ou x_k "rentre" entre ses bornes initiales. On supprime alors cette variable de la fonction $\phi_h(x)$ et on lui "rend" ses bornes initiales. On définit ainsi un nouveau programme (P'_{h+1}). Puis on recommence, jusqu'à ce que la fonction économique soit égale à fx, et on poursuit la maximisation jusqu'à l'optimum.
Il est possible que l'on atteigne une solution optimale alors qu'il reste des variables dans le terme de pénalisation. Il faut alors augmenter la valeur de δ jusqu'à ce que la solution ne soit plus optimale, et poursuivre la maximisation.

Remarque :

la fonction économique fx se réduit, dans la méthode des centres linéarisée, à la variable μ. Les variables contenues dans le terme $\phi_h(x)$ sont donc différentes de celles contenues dans le terme fx.
La remarque précédente est donc encore valable et, ici encore, il est inutile, en pratique, de modifier la valeur des bornes non vérifiées.
Ce procédé peut également être utilisé lors de l'addition de contraintes de centrage, en partant de la base optimale du programme linéaire précédent, ce qui évite d'utiliser la méthode duale-simpliciale.

ORGANIGRAMME DE LA METHODE



II.3 Utilisation d'une méthode de relaxation.

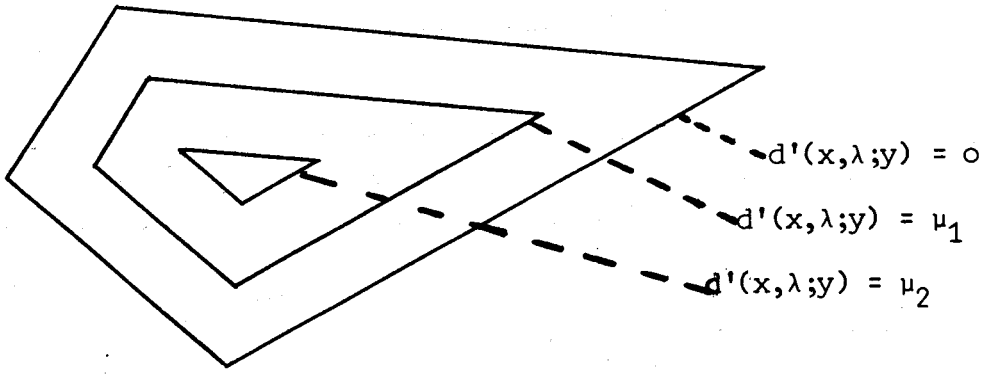
II.3.1 *Généralités*.

Rappelons que nous cherchons à maximiser la fonction
$d'(x,\lambda;y) = \min \{f'(x;y) - \lambda, g_i(x;y) | i \in L\}$.
Considérons le polyèdre linéaire P_o défini par

$$P_o \quad \begin{cases} f'(x;y) - \lambda \geq o \\ g'_i(x;y) \geq o \quad i \in L \end{cases}$$

Les équipotentielles de d' sont données par $d'(x;\lambda;y) = \mu$.
L'équipotentielle $d'(x,\lambda;y) = o$ correspond à la frontière du polyèdre linéaire P_o.
De même, pour un μ donné, l'équipotentielle définie par $d'(x,\lambda;y) = \mu$ correspond à la frontière du polyèdre :

$$P_\mu \quad \begin{cases} f'(x;y) - \lambda \geq \mu \\ g'_i(x;y) \geq \mu \quad i \in L \end{cases}$$



Posons $\bar{\mu} = d'(z,\lambda;y)$ la valeur optimale de la F-distance linéarisée, supposée finie.

PROPOSITION II.5. :

Une condition nécessaire et suffisante pour que le polyèdre P_μ soit vide est que $\mu > \bar{\mu}$.

PROPOSITION II.6. :

La valeur $\mu = o$ minore $\bar{\mu}$.

La démonstration de ces deux propriétés est immédiate.

II.3.2 *Algorithme général.*

On suppose qu'on dispose d'un point $\overset{o}{p}$ quelconque.
On se donne une valeur $\overset{o}{\mu}$ majorant $\bar{\mu}$; en général la valeur $\overset{o}{\mu} = \max \{f'(\overset{o}{p};y) - \lambda,\ g'_i(\overset{o}{p};y) \mid i \in L\}$ est une bonne valeur de départ.

Pour déterminer z, centre du polyèdre P_o, on applique l'algorithme suivant :

1) Faire $a = o$, $b = \overset{o}{\mu}$, $c = o$
   $r = o$.
2) A partir de $\overset{r}{p}$, déterminer un point $\overset{r+1}{p}$ intérieur au polyèdre P_c défini par

$$P_c \quad \begin{cases} f'(x;y) - \lambda \geq c \\ g'_i(x;y) \geq c \qquad i \in L \end{cases}$$

   Si $\exists\ \overset{r+1}{p}$, aller en 3), sinon aller en 4).
3) Soit $\overset{r+1}{\mu} = d'(\overset{r+1}{p},\lambda;y)$.
   Faire $a = \overset{r+1}{\mu}$ et aller en 5) avec r+1 au lieu de r.
4) Faire $b = c$ et aller en 5)
5) Faire $c = \dfrac{a + b}{2}$ et aller en 2).

On voit que les valeurs successives de a et b encadrent la valeur optimale $\bar{\mu}$, tout en se rapprochant l'une de l'autre. D'où le test d'arrêt de l'algorithme : on estime être arrivé à la solution z lorsque la différence b - a est inférieure à un Σ fixé à priori. Le point essentiel est la détermination, s'il existe, du point $\overset{r+1}{p}$ à partir de $\overset{r}{p}$.

II.3.3 *Algorithme partiel.*

A partir de $\overset{r}{p}$, on cherche, s'il existe, un point $\overset{r+1}{p}$ intérieur au polyèdre P_c. Nous utiliserons pour cela les méthodes de relaxation.

METHODE DE RELAXATION ([29]).

A une étape r nous considérons le polyèdre P_c.
Nous supposons que ce polyèdre a un intérieur non vide.

Soit un point q n'appartenant pas au polyèdre ; donc certaines des fonctions considérées sont négatives en q. Considérons celle dont la valeur en q a la plus grande valeur absolue ; soit π l'hyperplan correspondant. Soit q_1 la projection de q sur π.
Soit ρ tel que $0 < \rho \leq 2$. Le point

$$q' = q + \rho(q_1 - q)$$

appartient au segment $]q,q_2]$, q_2 étant le symétrique de q par rapport à π.

Si q' appartient au polyèdre, on a trouvé le point $\overset{r+1}{p}$ cherché.
Sinon, on fait q = q' et on itère le processus.

- Si $0 < \rho < 2$, le processus converge, mais non nécessairement en un nombre fini d'itérations.

- Si $\rho = 2$, le processus converge en un nombre fini d'itérations.

Nous avons donc choisi $\rho = 2$, c'est-à-dire que $q' = q_2$, symétrique de q par rapport à π (méthode de réflexion).

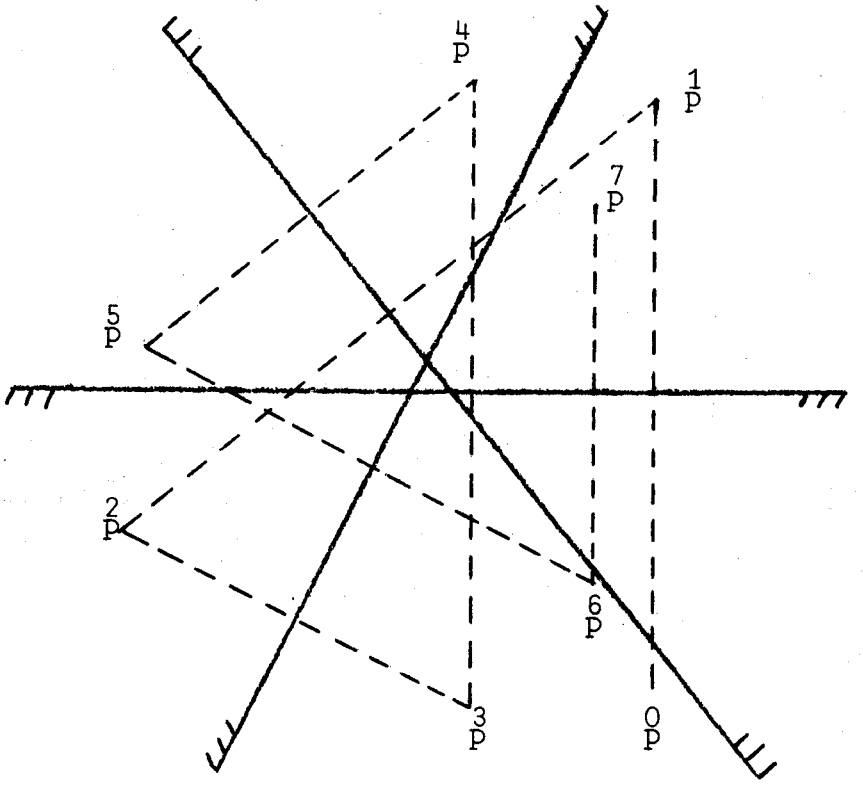
METHODE DE RELAXATION GENERALISEE ([28])

Dans la méthode de relaxation, on cherche le point q' dans la direction du gradient de l'une seulement des contraintes non vérifiées. Dans la méthode de relaxation généralisée, la direction considérée est une combinaison linéaire des gradients de toutes les contraintes non vérifiées.

ORGANISATION PRATIQUE DES CALCULS

A chaque étape il nous faut déterminer si le polyèdre considéré est vide ou non. Le test de vacuité que nous avons utilisé est le suivant : on applique à partir du point $\overset{r}{p}$ la méthode de relaxation, et on estime que le polyèdre est vide si le nombre d'itérations dépasse un nombre N fixé d'avance.

Or, il est possible qu'à une étape r donnée, le nombre d'itérations nécessaire pour obtenir un point $\overset{r+1}{p}$ à partir de $\overset{r}{p}$ dépasse N, bien que le polyèdre ne soit pas vide. En particulier, la méthode de relaxation converge lentement lorsque la "distance" du point au polyèdre est grande par rapport à la "taille" du polyèdre :



Pour éviter cet inconvénient :

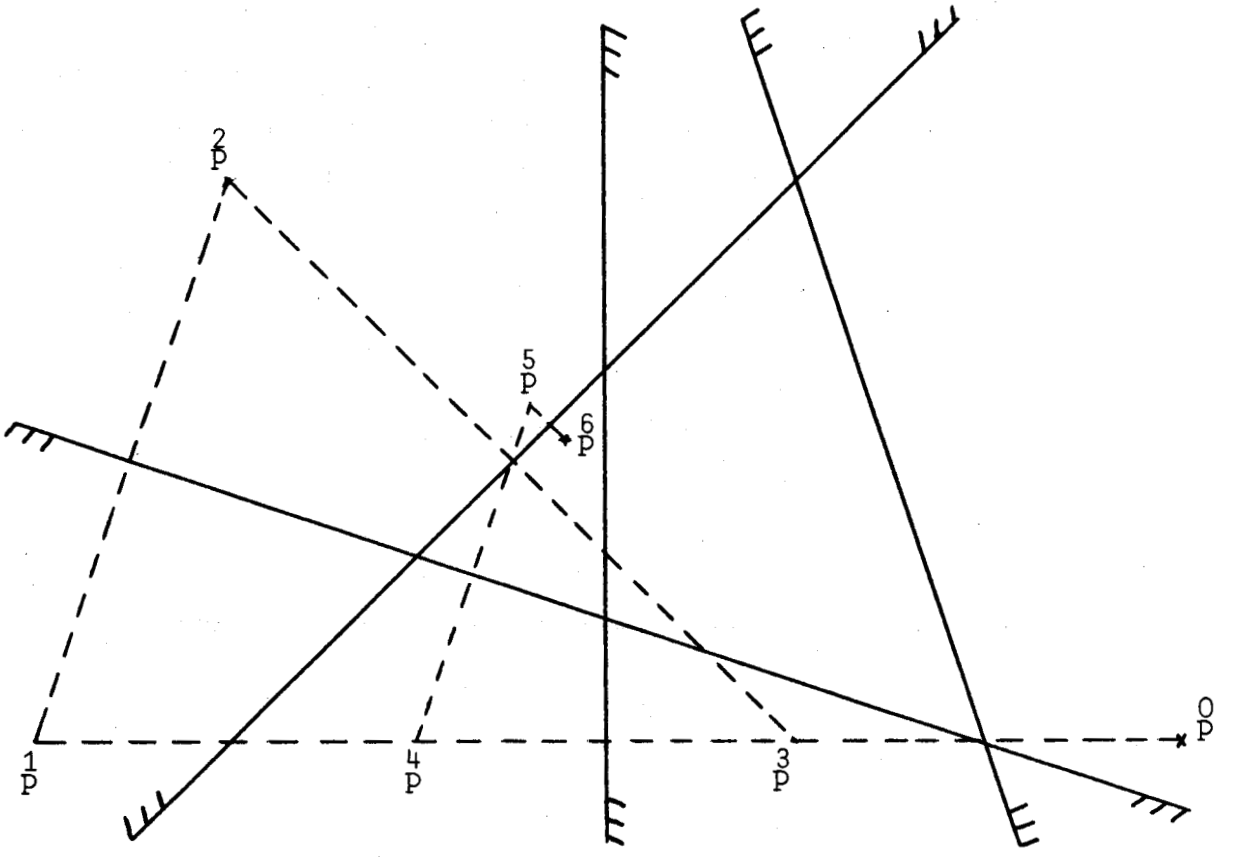
- on donne à N une valeur importante ;

- chaque fois que la valeur de a augmente de $\Delta\mu$, on donne arbitrairement à b le même accroissement $\Delta\mu$; intuitivement, ceci conduit à "essayer plusieurs fois" une valeur de c qui n'a "pas marché", le point $\overset{r}{p}$ s'étant entre temps "rapproché" du polyèdre.

II.3.4 *Expériences numériques.*

1er essai : Trouver le centre d'un triangle dans $\mathbb{R}^2$, les variables étant assujetties à des bornes.

Il s'agit d'un essai préliminaire très simple pour visualiser le comportement de l'algorithme.



1	3
-3	-1
1	-1

3
-9
-1

matrice des contraintes

second membre

Bornes inférieures : (-10,-10)
Bornes supérieures : (1, 10)
Point de départ : (4, 0)
Valeur maximale de d' : 0,5402.

La méthode de relaxation simple (méthode de réflexion) résout facilement le problème. Après plusieurs essais, nous avons constaté qu'il suffit de prendre pour N la valeur $N = 10$. Dans ces conditions, en fixant Σ à 10^{-9}, nous obtenons le centre au bout de 359 itérations. La figure II.1 montre les valeurs prises successivement par a et b et nous voyons qu'elles encadrent correctement la valeur optimale de d'.

2eme essai : Trouver le centre d'un polyèdre défini par 5 contraintes dans $\mathbb{R}^9$, les variables étant assujetties à des bornes. Ce problème a été obtenu lors d'une linéarisation d'un tronçon dans le traitement du problème n°4 (voir annexe A1).

0	0	0	-0,4228	-0,4379	-0,4530	-0,4530	-0,4681	0	-626,48
-0,1738	-0,1047	-100	0,1002	0,1002	0,1002	0	0	0	-101,57
0,1950	0,0976	100	0	0	0	0,1101	0,1101	0	136,70
-0,4029	0,5425	-100	0	0	0	0	0	0,2281	-4,7899
0,4424	-0,6009	-100	0	0	0	0	0	0	-57,524

Matrice des contraintes — second membre

Bornes inférieures : 340,340,0,0,0,0,0,0,-1000.
Bornes supérieures : 420,420,0.5236,100,100,800,300,100,1000
Point de départ : 340.5, 419.5,0.5,100,100,800,300,90,191.175
Valeur maximale de d' : 0,3573.

De nombreux essais ont été effectués avec la méthode de relaxation simple (méthode de réflexion). Ces essais ont conduit à donner à N des valeurs importantes ; nous avons été jusqu'à N = 5000. La figure II.2 montre, dans le cas où N = 5000, les valeurs prises successivement par a et b ; comme ces valeurs n'encadrent pas correctement la valeur optimale de d', nous voyons que cette valeur de N est encore insuffisante. Mais comme le volume des calculs croît considérablement en fonction de N, le temps de calcul devient prohibitif (de l'ordre de plusieurs dizaines de minutes). La méthode résout donc très difficilement le problème.

Nous avons également sur ce problème effectué quelques essais avec la méthode de relaxation généralisée, en prenant pour direction de progression la somme des gradients de toutes les contraintes non vérifiées en $\overset{r}{p}$. Ces essais ont donné des résultats encore moins intéressants et des temps de calcul encore plus élevés.

CONCLUSION

La vitesse de convergence de la méthode peut être très faible. Il semble en particulier qu'elle dépende fortement de la forme du polyèdre considéré. L'application de la méthode à un problème concret dans $\mathbb{R}^9$ a donné des résultats peu intéressants, le temps de calcul étant de l'ordre de plusieurs centaines de fois le temps nécessaire pour résoudre le problème à l'aide de la méthode simpliciale.

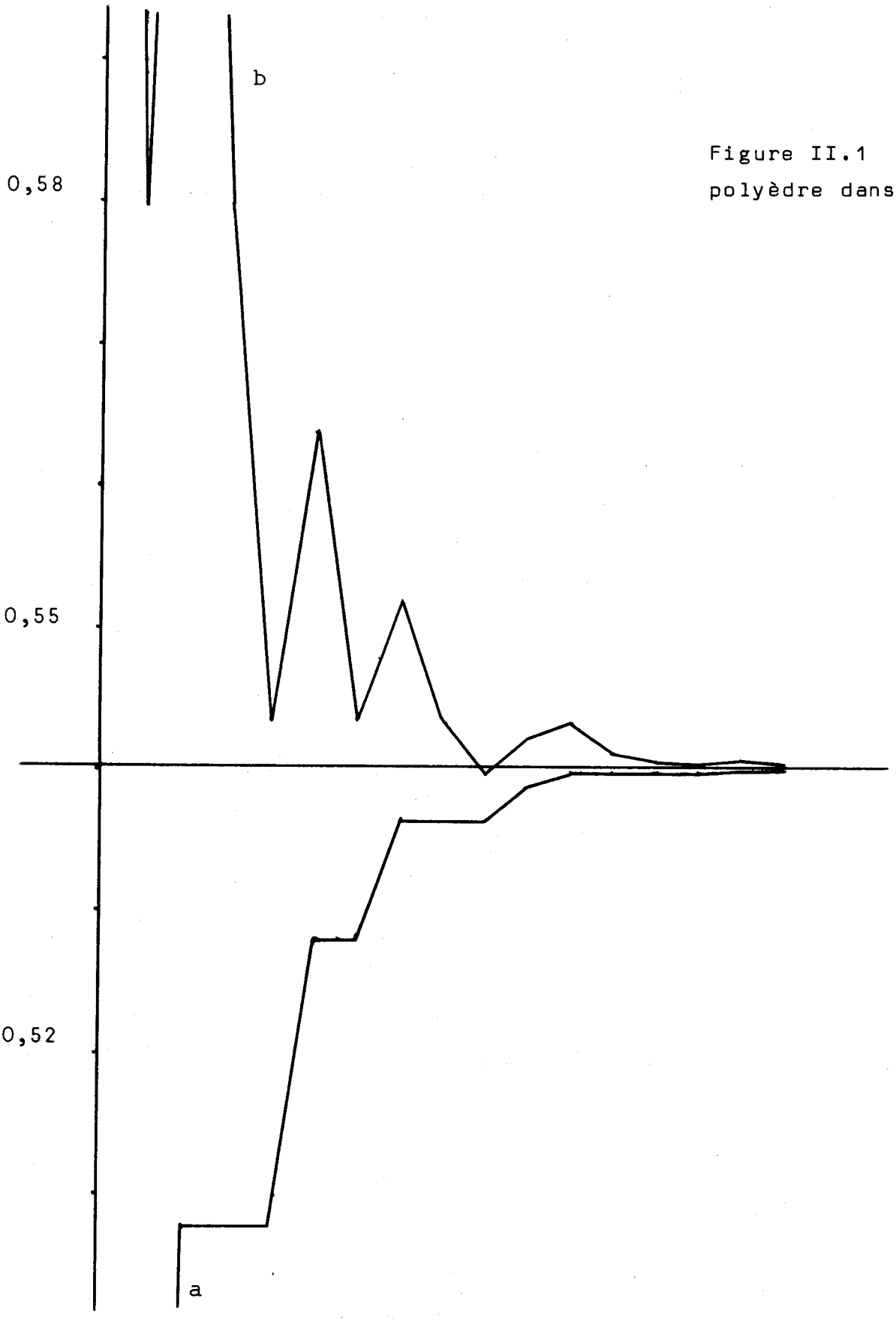


Figure II.1
polyèdre dans $\mathbb{R}^2$.

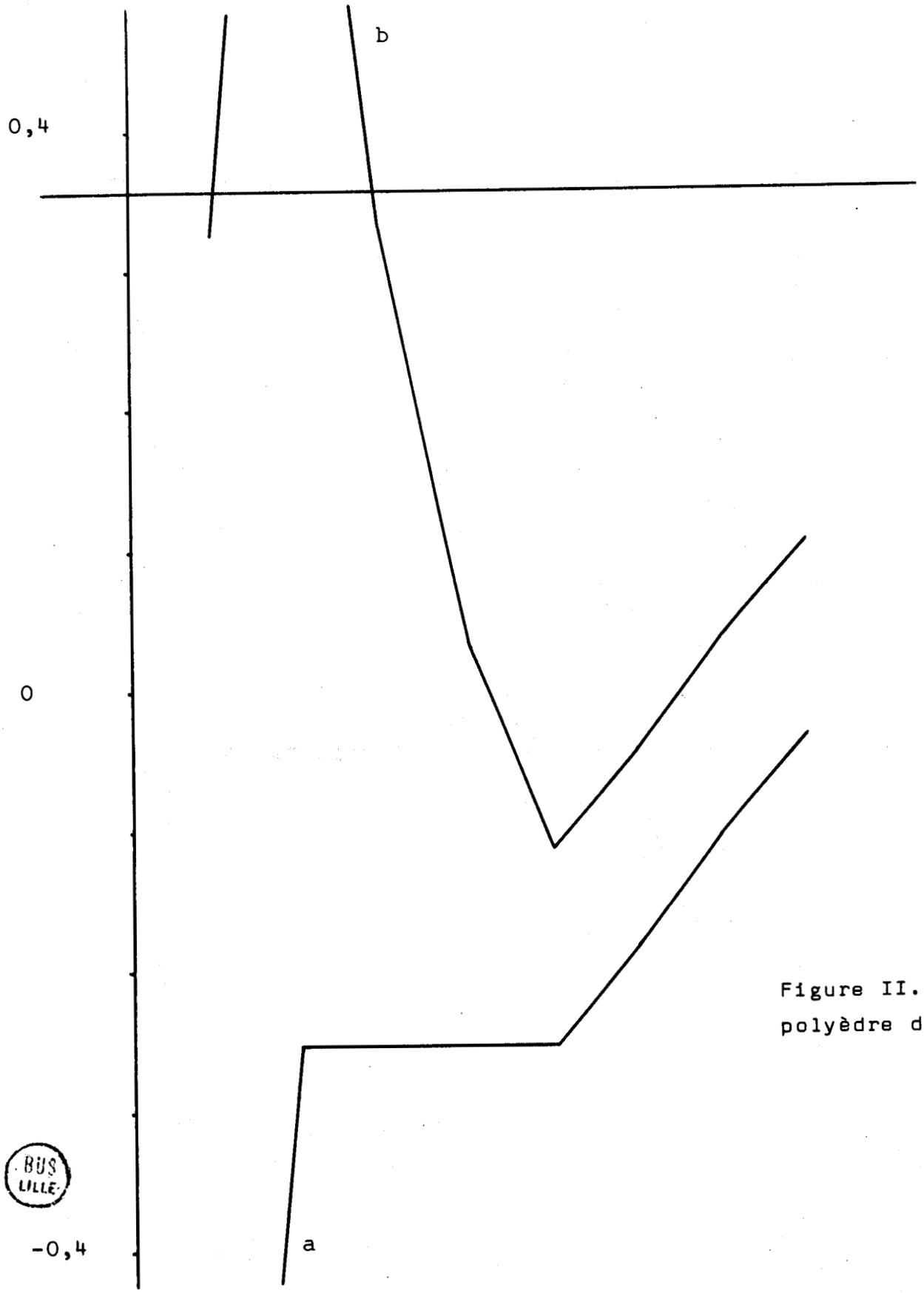


Figure II.2
polyèdre dans $\mathbb{R}^9$

II.3.5 *La méthode de sous-gradients.*

Dans un article récent ([19]), HELD, WOLFE et CROWDER ont proposé un algorithme analogue à l'algorithme précédent.
Soit $G(\overset{r}{p})$ l'ensemble des gradients des fonctions qui sont égales en $\overset{r}{p}$ à $d'(\overset{r}{p},\lambda;y)$.
En général, $G(\overset{r}{p})$ est réduit à un seul élément ; dans ce cas, d'est différentiable en $\overset{r}{p}$ et cet élément est le gradient $\nabla d'(\overset{r}{p},\lambda;y)$.
Sinon, d' n'est pas différentiable en $\overset{r}{p}$. Dans ce cas, on choisit un sous-gradient extrême $s_i \in G(\overset{r}{p})$.
Donc dans les deux cas, on considère un vecteur unique $s(\overset{r}{p}) \in G(\overset{r}{p})$.
Le point de départ $\overset{o}{p}$ étant donné, on considère une suite $\{t_r\}$ de scalaires positifs, et on définit la suite $\{\overset{r}{p}\}$ par

$$\overset{r+1}{p} = \overset{r}{p} + t_r s(\overset{r}{p}) \qquad r = 1,2,\ldots$$

$d'(\overset{r}{p},\lambda;y)$ converge vers son maximum $\bar{\mu}$ si

$$t_r \to o, \quad \sum_{r=o}^{\infty} t_r = \infty.$$

On choisit

$$t_r = \rho_r \frac{\mu - d'(\overset{r}{p},\lambda;y)}{||s(\overset{r}{p})||^2} \quad \text{avec } \mu < \bar{\mu}.$$

et $\forall r$, $\varepsilon < \rho_r \leq 2$ pour un $\varepsilon > o$ fixé.

Cet algorithme est le même que le précédent. Le cas particulier de la méthode "des symétriques" (méthode de réflexion) est le cas où

$$\rho_r = 2 \ \forall r.$$

Remarque :

Si au lieu de prendre $\mu < \bar{\mu}$ on prend $\mu > \bar{\mu}$ ce qui est nécessaire en pratique, les auteurs considèrent une suite ρ_r tendant vers o ; on a alors toujours $t_r \to o$ (mais on n'a plus $\Sigma t_r = \infty$). L'algorithme est alors différent du précédent et semble donner, dans certains cas, des résultats intéressants, bien qu'il soit plus lent que la résolution du problème à l'aide de la méthode simpliciale.

II.4. Méthodes de rangement d'une matrice sous forme compacte.

II.4.1 *Généralités.*

L'avantage des méthodes que nous venons d'exposer est de permettre le rangement des matrices considérées au moindre encombrement mémoire, en utilisant le creux de ces matrices.
La méthode simpliciale, telle que nous l'avons exposée en II.2.7, nécessite l'implantation en mémoire de la matrice A et de l'inverse de base $(A^I)^{-1}$. La matrice A, n'étant pas modifiée au cours de la résolution, peut être stockée sous forme condensée.
La matrice $(A^I)^{-1}$ sera factorisée et chacun des facteurs également stocké sous forme condensée.
Si nous utilisons les méthodes de relaxation, la seule matrice à stocker est la matrice A, et comme elle n'est pas modifiée au cours des calculs, on peut la ranger de manière condensée.

Nous allons donc examiner rapidement divers procédés de rangement d'une matrice sous forme condensée.

II.4.2 *Techniques de rangement et schémas d'indiçage.*

Les principales méthodes sont répertoriées dans [30].

1ère méthode : schéma booléen.

On définit une matrice boolèenne B telle que

$$b_i^j = \begin{cases} 1 \text{ si } a_i^j \neq o \\ o \text{ si } a_i^j = o \end{cases}$$

On range en mémoire la matrice B, et les éléments non nuls de A en lignes ou en colonnes suivant les cas.
La valeur d'un élément de A est trouvée en comptant les 1 dans la matrice B. Pour obtenir un accès plus rapide, on peut également ranger en mémoire un vecteur dont les éléments donnent l'adresse du premier élément non nul de chaque ligne (ou de chaque colonne).

2e méthode : schéma par adresses.

On range en mémoire les éléments non nuls de A, et on définit une matrice B telle que

$$b_i^j = \begin{cases} \text{adresse de } a_i^j \text{ si } a_i^j \neq o \\ o \text{ si } a_i^j = o \end{cases}$$

Il est clair que l'accès est plus rapide que dans la méthode précédente, mais que l'encombrement mémoire est beaucoup plus important.

3e méthode : schéma lignes-colonnes.

Pour chaque élément non nul de A, on conserve en mémoire la valeur de l'élément, son indice de ligne et son indice de colonne.

Remarquons que si les éléments non nuls de A sont rangés ligne par ligne (respectivement colonne par colonne) il est inutile de conserver l'indice de ligne (respectivement de colonne) de chaque élément ; il suffit de conserver le rang du premier élément non nul de chaque ligne (respectivement de chaque colonne).

Ce procédé peut conduire à tester fréquemment la valeur des indices, mais il est particulièrement avantageux du point de vue encombrement mémoire.

4e méthode : schéma en liste.

Comme dans la méthode précédente, on conserve en mémoire la valeur des éléments non nuls de A, leur indice de ligne et leur indice de colonne. De plus, on crée une structure de liste en associant à chacun des éléments un chaînon dont la valeur donne l'adresse de l'élément suivant de la liste. L'intérêt de ce procédé est de permettre la modification, sans transfert d'éléments, d'une matrice rangée sous forme condensée, par insertion ou suppression d'un élément dans la liste.
Comme dans la méthode 3, ce procédé conduit à des tests fréquents sur les indices. De plus, si l'on veut pouvoir récupérer la place laissée libre par la suppression d'un élément, il faut enregistrer une table des adresses inoccupées.

Remarque :

Il est clair que si la matrice A posséde une structure particulière, on peut définir des méthodes spécialement adaptées à la forme de la matrice, qui permettent un gain en temps d'accès et en encombrement mémoire.

II.4.3 *Technique utilisées dans la méthode simpliciale.*

Nous devons ranger la matrice A en tenant compte des particularités de la méthode utilisée. En effet, la création de la matrice A s'effectue ligne par ligne, puisque les contraintes sont linéarisées l'une après l'autre. Par contre, l'accès s'effectue colonne par colonne, puisque lors d'un changement de base les calculs utilisent la colonne A^s.

Nous devons donc utiliser une méthode de rangement qui permette ces deux opérations avec facilité et rapidité, tout en minimisant l'encombrement mémoire.

La méthode retenue est une combinaison d'un schéma lignes-colonnes et de schémas en liste : on conserve en mémoire la valeur des éléments non nuls de A rangés ligne par ligne, leur indice de ligne et leur indice de colonne. De plus, on crée une structure de liste par colonne en associant à chacun des éléments un chaînon dont la valeur donne l'adresse de l'élément précédent de la colonne, et en conservant, pour chaque colonne, la tête de liste dans un tableau annexe.

La création de A s'effectuant ligne par ligne, il suffit de ranger les éléments de manière consécutive, au fur et à mesure de leur calcul. En même temps, on crée les n listes progressivement, en donnant au chaînon de chaque nouvel élément créé la valeur de l'adresse du précédent élément créé dans la colonne correspondante, et en conservant comme tête de liste l'adresse du dernier élément créé.

L'accès, dans la méthode simpliciale, se faisant par colonnes, il suffit, si l'on veut obtenir la colonne A^s, de parcourir la liste s, en partant de la tête de liste.

On peut remarquer que, dans ce schéma, l'adjonction de contraintes de centrage s'effectue de la même manière et ne pose donc aucun problème.

CHAPITRE III

MAXIMISATION DE LA F-DISTANCE SUR UN SEGMENT

III.1 Introduction :

Un point essentiel de la méthode des centres linéarisée est la maximisation de la F-distance sur un segment. Ce sous-problème doit être résolu aussi bien dans l'algorithme partiel (I.4) que dans l'algorithme de centrage (I.5) et se présente donc fréquemment.

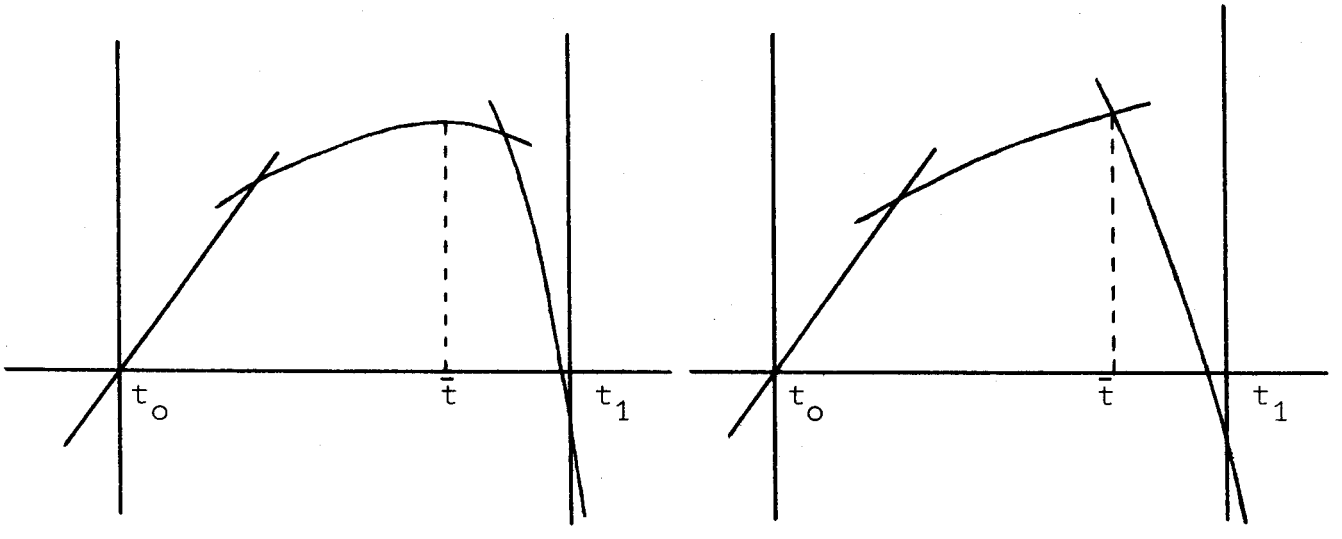
En notant $[t_o, t_1]$ le segment considéré, la fonction à maximiser est donc :

$$h(\theta) = d(t_o + \theta(t_1 - t_o), E(\lambda)) \qquad \theta \in [0,1].$$

et on doit trouver $\bar{\theta}$ tel que $h(\bar{\theta}) = \max_{o \leq \theta \leq 1} \{h(\theta)\}$.

La fonction h est une enveloppe inférieure de fonctions concaves, chacune de ces fonctions étant la restriction au segment $[t_o, t_1]$ d'une des contraintes du problème (P). Elle est donc elle-même concave.

Des tabulations de d sur $[t_o, t_1]$ montrent que cette fonction possède l'une des deux formes suivantes :



(1) (2)

La forme 2 se produisant beaucoup plus souvent que la forme 1 : dans la plupart des cas le maximum se trouve à l'intersection de deux contraintes.

Il existe dans la littérature deux classes principales de méthodes de maximisation d'une fonction concave sur un segment : les méthodes de fractionnement (ou ponctuelles) et les méthodes d'interpolation. Après un bref rappel de ces méthodes, nous proposons une méthode d'interpolation polygonale bien adaptée à la forme particulière de la fonction.

III.2 Méthodes de fractionnement.

III.2.1 *Généralités.*

Les méthodes de fractionnement sont basées sur la réduction du segment $[t_o, t_1]$. A chaque étape, elles ne nécessitent que le calcul des valeurs de la fonction $h(\theta)$ en des points déterminés d'après les résultats des étapes précédentes, et sont donc faciles à mettre en oeuvre ; elles convergent sous la seule hypothèse que h est unimodale sur l'intervalle, ce qui est le cas si elle est concave.

La réduction de l'intervalle s'effectue sans tenir compte de la forme particulière de la fonction ; la vitesse de convergence de ces méthodes est donc connue à priori, ce qui fait qu'elles sont sûres ; elles sont cependant en général assez lentes.

III.2.2 *Dichotomie.*

La longueur de l'intervalle de confiance est divisée par 2 à chaque étape.

L'intervalle de départ pour θ est $[0,1]$. On fait donc au départ

$$a_1 = o,\ b_1 = 1.$$

1) Faire $i = 1$

2) Calculer $c_i = \dfrac{a_i + b_i}{2}$

Si $h(c_i) < h(a_i)$, prendre $a_{i+1} = a_i$ et $b_{i+1} = c_i$.
Aller en 2) avec $i = i+1$.

Si $h(c_i) < h(b_i)$, prendre $a_{i+1} = c_i$ et $b_{i+1} = b_i$.
Aller en 2) avec $i = i+1$.

Sinon :

3) Calculer $d_i = \dfrac{a_i + c_i}{2}$

Si $h(c_i) < h(d_i)$, prendre $a_{i+1} = a_i$, $b_{i+1} = c_i$ et $c_{i+1} = d_i$.
Aller en 3) avec $i = i+1$.

Sinon :

4) Calculer $e_i = \dfrac{b_i + c_i}{2}$

Si $h(c_i) < h(e_i)$, prendre $a_{i+1} = c_i$, $b_{i+1} = b_i$ et $c_{i+1} = e_i$.
Aller en 3) avec $i = i+1$.

Sinon :

5) Prendre $a_{i+1} = d_i$, $b_{i+1} = e_i$, $c_{i+1} = c_i$,
et aller en 3) avec $i = i+1$.

On arrête le processus lorsque le segment a été réduit à une fraction donnée de la norme du segment d'origine.

III.2.3 *Section dorée* :

Cette méthode est basée sur la propriété suivante :
Si dans l'intervalle $[a_i, b_i]$ on connaît la valeur de h en 2 points c_i et d_i tels que

$$a_i < c_i < d_i < b_i,$$

Si $h(c_i) < h(d_i)$, le maximum est sur $[c_i, b_i]$
Si $h(c_i) > h(d_i)$, le maximum est sur $[a_i, d_i]$

On détermine les points c_i et d_i de manière que la taille de l'intervalle contenant le maximum diminue d'un facteur r constant à chaque étape.

On doit donc avoir :

$$\frac{d_i - a_i}{b_i - a_i} = \frac{b_i - c_i}{b_i - a_i} = r \quad \text{et} \quad c_i - a_i = b_i - d_i.$$

Supposons par exemple $h(d_i) < h(c_i)$ (le cas contraire est symétrique).
Donc $b_{i+1} = d_i$, $a_{i+1} = a_i$, et on prend $d_{i+1} = c_i$.

$$\frac{d_{i+1} - a_{i+1}}{b_{i+1} - a_{i+1}} = r \quad \text{par définition de } r$$

D'où

$$\frac{c_i - a_i}{d_i - a_i} = r.$$

Or $c_i - a_i = b_i - d_i = b_i - a_i - (d_i - a_i)$

$$\Longrightarrow \frac{c_i - a_i}{d_i - a_i} = \frac{b_i - a_i}{d_i - a_i} - 1 = \frac{1}{r} - 1 = r.$$

Donc $r^2 + r - 1 = 0$.

La racine positive de cette équation est $r = \dfrac{\sqrt{5} - 1}{2}$

D'où l'algorithme :

1) $a_1 = o$, $b_1 = 1$. Faire $i = 1$.
   $c_1 = a_1 + (1-r)(b_1 - a_1)$, $d_1 = b_1 - (1 - r)(b_1 - a_1)$

2) Si $h(d_i) < h(c_i)$, prendre $a_{i+1} = a_i$, $b_{i+1} = d_i$,
   $d_{i+1} = c_i$.

Calculer $c_{i+1} = a_{i+1} + (1 - r)(b_{i+1} - a_{i+1})$
Aller en 2) avec $i = i + 1$.

Sinon :

3) Prendre $a_{i+1} = c_i$, $b_{i+1} = b_i$, $c_{i+1} = d_i$.
Calculer $d_{i+1} = b_{i+1} - (1 - r)(b_{i+1} - a_{i+1})$.
Aller en 2) avec $i = i+1$.

Après n calculs de fonction, on a

$$b_n - a_n = (b_1 - a_1)r^{n-1}.$$

On peut faire un test d'arrêt sur la norme de $[a_i, b_i]$.

III.2.4 *Méthode basée sur les nombres de Fibonacci*.

Comme dans la section dorée, on détermine dans l'intervalle $[a_i, b_i]$ 2 points c_i et d_i à chaque étape.

Supposons que l'intervalle final est de longueur unité. La meilleure méthode est celle pour laquelle, étant donné un certain nombre de calculs de fonction, l'intervalle initial est le plus grand possible. Soit donc L_n la longueur du plus grand intervalle pouvant être réduit à un intervalle unité après n calculs de fonction.

Considérons la première subdivision : $a_1 < c_1 < d_1 < b_1$.

Si le maximum est sur $[a_1, c_1]$, on peut encore déterminer $n - 2$ points sur cet intervalle pour terminer la maximisation : il faut donc que $c_1 - a_1 \leq L_{n-2}$.

Si le maximum est sur $[c_1, b_1]$, on peut de même déterminer n - 2 points supplémentaires sur cet intervalle, mais on peut encore de plus utiliser d_1 ; on peut donc utiliser n - 1 points pour terminer la maximisation : il faut que $b_1 - c_1 \leq L_{n-1}$.
En faisant la somme de ces 2 inégalités on obtient

$$b - a = L_n \leq L_{n-1} + L_{n-2}.$$

On a $L_o = L_1 = 1$ car on ne peut réduire l'intervalle avec moins de 2 calculs de fonction. D'autre part, comme on veut que L_n soit le plus grand possible.:

$$L_n = L_{n-1} + L_{n-2}.$$

La solution de cette équation de récurrence donne les nombres de Fibonacci :

$$F_i = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^{i+1} - \left(\frac{1 - \sqrt{5}}{2} \right)^{i+1} \right]$$

D'où l'algorithme :

1) $a_1 = o$, $b_1 = 1$. Faire $i = 1$.

$$c_1 = \frac{F_{n-2}}{F_n} (b_1 - a_1) + a_1, \quad d_1 = \frac{F_{n-1}}{F_n} (b_1 - a_1) + a_1.$$

2) Si $h(d_i) < h(c_i)$, Prendre $a_{i+1} = a_i$, $b_{i+1} = d_i$, $d_{i+1} = c_i$.

Calculer $c_{i+1} = \frac{F_{n-i-2}}{F_{n-i}} (b_{i+1} - a_{i+1}) + a_{i+1}$.

Aller en 2) avec $i = i + 1$.

Sinon :

3) Prendre $a_{i+1} = c_i$, $b_{i+1} = b_i$, $c_{i+1} = d_i$.

Calculer $d_{i+1} = \frac{F_{n-i-1}}{F_{n-i}} (b_{i+1} - a_{i+1}) + a_{i+1}$.

Aller en 2) avec i = i + 1.

Après n calculs de fonction, on a

$$b_n - a_n = \prod_{i=1}^{n} \frac{F_{n-i}}{F_{n-i+1}} (b_1 - a_1) = \frac{b_1 - a_1}{F_n} .$$

Le test d'arrêt s'effectue sur le nombre de calculs de fonction fixé à l'avance.

III.3 Méthodes d'interpolation.

III.3.1 *Interpolation polynomiale.*

Connaissant les valeurs prises par la fonction h en n+1 points, on interpole la fonction par un polynome de degré n. On recherche le maximum de ce polynome, et on remplace un des n+1 points par le point obtenu ; et ainsi de suite.

En pratique, on fait l'interpolation par un polynome de degré 2 en général, car on sait facilement en déterminer le maximum.

Il faut toutefois prendre certaines précautions pour assurer la convergence.

Il est clair que ce type de méthodes ne peut donner de bons résultats que si la fonction à maximiser est, au voisinage de l'optimum, bien représentée par un polynome de degré 2. Or, si nous considérons la forme (2) de la F-distance (III.1), nous voyons que, dans la plupart des cas, le maximum se trouve en un point où la fonction n'est pas différentiable. L'interpolation polynomiale est donc très mal adaptée à la maximisation d'une F-distance.

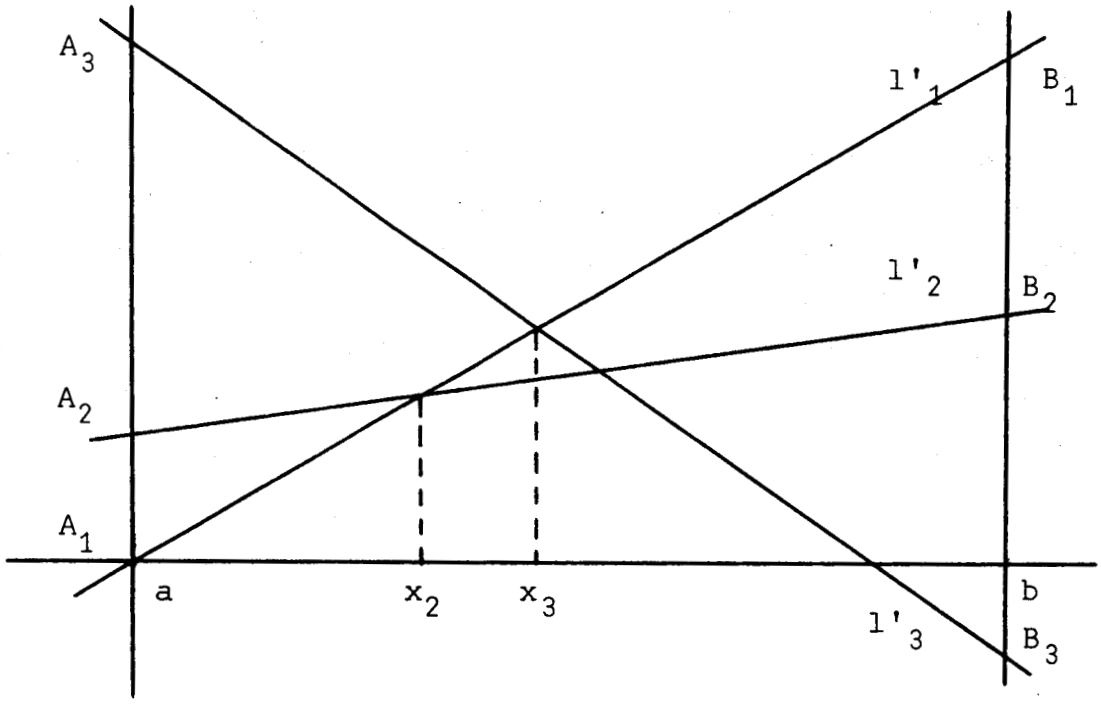
III.3.2 *Interpolation polygonale* :

La F-distance, n'étant pas partout différentiable, ne peut être approximée de manière valable par un polynome d'interpolation. Par contre, chacune des fonctions dont elle est l'enveloppe inférieure est différentiable et peut donc être interpolée par un polynome.

Pour obtenir un procédé de maximisation conduisant à un nombre réduit de calculs, nous avons utilisé pour chacune des fonctions une interpolation linéaire, ce qui revient à interpoler la F-distance par un polygone.

Dans une première phase, on effectue une réduction de l'intervalle de confiance. Cette réduction peut s'effectuer par la détermination de points c, d, et e, choisis, comme dans l'algorithme de dichotomie, au milieu des segments ou sous-segments considérés. Toutefois il est préférable, chaque fois que c'est possible, d'utiliser le polygone d'interpolation pour déterminer les points c, d, et e, ou certains de ces points.

Considérons le segment [a,b]. On détermine le point c qui maximise l'énveloppe inférieure des droites d'interpolation, de la façon suivante :



Soient A_i et B_i les valeurs prises par la contrainte (ou l'équipotentielle économique) l_i en a et b.

On considère au départ la fonction l_r qui donne le minimum des l_i en a, c'est-à-dire la valeur de la F-distance en a. Soit $x_r = a$.

1) Calculer les abscisses x_j des points d'intersection de la corde l'_r avec les autres cordes l'_j, $j \neq r$;

$$x_j = \frac{A_r b - a B_r - A_j b + a B_j}{A_r - B_r - A_j + B_j} \quad , \; j \neq r.$$

Déterminer la première intersection rencontrée lorsqu'on se déplace le long de la corde l'_r, c'est-à-dire déterminer s tel que

$$x_s = \min \{x_j \mid x_j > x_r\}.$$

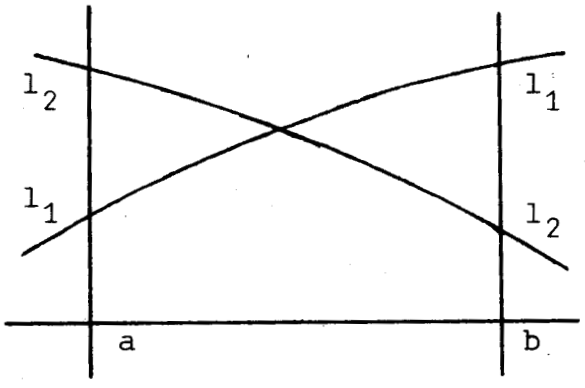
2) On a donc déterminé une nouvelle fonction l_s. Si la corde l'_s a une pente positive, on peut encore améliorer, en "se déplaçant" le long de cette corde, la valeur de la F-distance.
Donc :
Si $B_s > A_s$, aller en 1) avec s au lieu de r.
Sinon, on est à l'optimum : faire $c = x_s$.

Si $c \in]a,b[$, il est intéressant d'utiliser ce point pour la réduction de l'intervalle au lieu d'utiliser le point milieu de [a,b].

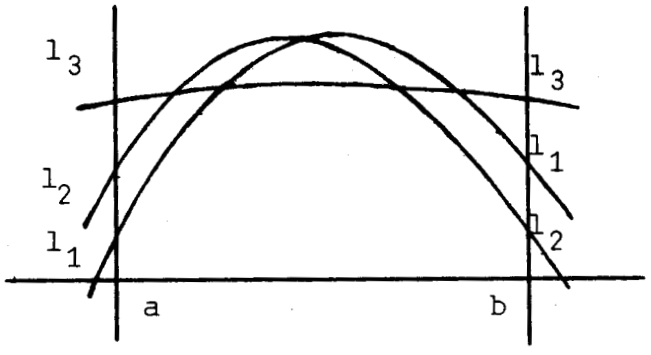
Remarque :

On peut effectuer une "interpolation partielle" en ne conservant, à chaque réduction d'intervalle, que les contraintes ayant les valeurs les plus faibles aux deux extrémités de l'intervalle, ce qui, moyennant certaines précautions, conduit à une réduction du volume des calculs.

Si la F-distance a la forme (2) de III.1., on obtient, au bout d'un certain nombre de réductions de l'intervalle, un nouvel intervalle tel qu'à ses extrémités les deux contraintes ayant les valeurs les plus faibles soient les mêmes, selon le schéma suivant :



On fait alors l'hypothèse que le maximum de la F-distance se trouve à l'intersection de ces deux contraintes. Cette hypothèse n'est pas nécessairement vérifiée, comme le montre le schéma suivant :



Toutefois elle est vérifiée si la courbure des fonctions n'est pas très accentuée, ce qui est souvent le cas en pratique: Il faudra cependant prendre certaines précautions pour ne pas aboutir à une solution erronée.

L'hypothèse ci-dessus étant faite, il suffit de rechercher le point d'intersection des deux contraintes, c'est-à-dire le zéro de leur différence. Cette recherche s'effectue très simplement par une méthode d'interpolation linéaire (méthode de la corde).

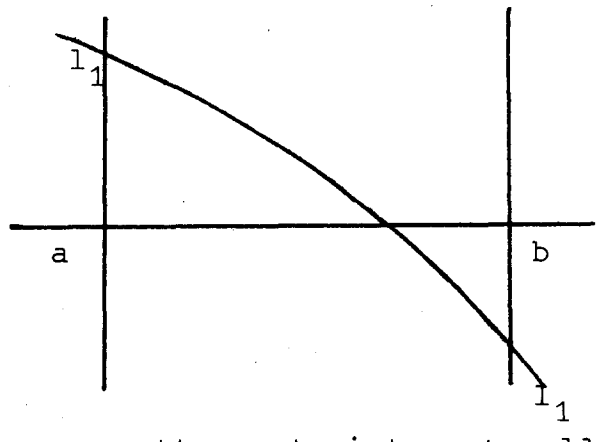
III.4. Recherche d'un point frontière :

Dans l'algorithme de centrage (I.5.) il est nécessaire de déterminer, non seulement le maximum de la F-distance sur un segment, mais aussi le point où le segment rencontre la frontière du tronçon, c'est-à-dire le zéro de la F-distance sur le segment.

On peut utiliser pour trouver ce point des méthodes analogues aux méthodes précédentes, en particulier la dichotomie.

On peut également employer un procédé semblable à celui de III.3.2 :

On réduit l'intervalle par dichotomie jusqu'à ce qu'à ses extrémités, la contrainte dont la valeur donne celle de la F-distance soit la même, selon le schéma suivant :



On estime alors que cette contrainte est celle dont l'intersection avec le segment est le point où la F-distance s'annule ; il suffit alors de rechercher le zéro de cette fonction par interpolation linéaire.

III.5. Performances comparées de deux algorithmes :

Le procédé d'interpolation polygonale de III.3.2. a fait l'objet d'un code, réalisé de la manière suivante :
La 1e phase (réduction de l'intervalle) est effectuée par dichotomie.
La 2e phase (interpolation) est effectuée en ne calculant que 2 contraintes au lieu de m+1. Dans les cas où l'interpolation linéaire ne peut donner de résultats (par exemple lorsqu'on a la forme (1) de III.1), le code applique automatiquement l'algorithme de dichotomie.

Les comparaisons ont été faites entre ce procédé et l'algorithme de dichotomie.

Nombre de calculs de la F-distance dans la résolution du problème n°5, à la 12e troncature (valeur de fonction économique 1501), durant chaque maximisation unidimensionnelle :

N° de maximisation unidimensionnelle	1e phase	2e phase
1	4	13
2	17	2
3	11	9
4	14	10
5	7	12
6	7	18
7	7	9
8	5	14
9	5	10
10	5	13
11	9	5
12	10	12
13	4	14

Pour une maximisation donnée, le nombre total de calculs de F-distance dans les deux phases est nettement inférieur au nombre de calculs nécessité par la dichotomie, qui est de 51 pour chaque maximisation. De plus, les calculs dans la 2e phase ne nécessitent que le calcul de 2 fonctions au lieu de 21. Le temps de calcul nécessaire pour déterminer le maximum et le zéro de la F-distance au cours de l'optimisation du problème n°5, est en moyenne divisé par 3 environ par rapport à l'algorithme de dichotomie.

Soit T_i le temps de résolution total nécessaire pour optimiser le problème n°5 en obtenant la valeur 1499,24 pour la fonction économique avec l'interpolation ; et soit T_d le temps de résolution nécessaire pour obtenir la même valeur avec la dichotomie.

On a :

$$T_i = 0{,}59\ T_d.$$

CHAPITRE IV

PRISE EN COMPTE

DES CONTRAINTES EN EGALITES

IV.1 Introduction :

Considérons le programme mathématique

$$(P) \quad \begin{cases} \text{Max } f(x) \\ a(x) \geqslant o \\ b(x) = o \end{cases} \qquad x \in \mathbb{R}^n$$

avec $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $a : \mathbb{R}^n \longrightarrow \mathbb{R}^p$, $b : \mathbb{R}^n \longrightarrow \mathbb{R}^q$.

Les contraintes $b(x) = o$ sont des contraintes en égalités.
Le domaine des solutions réalisables d'un tel problème est vide, ce qui interdit sa résolution par des méthodes intérieures. En particulier, la méthode des centres exige que l'intérieur du domaine soit non vide.

Pour résoudre (P) par une telle méthode, on peut chercher à définir un programme mathématique (P'), comportant uniquement des contraintes en inégalité, dont le domaine ait un intérieur non vide, et qui soit au moins quasi-équivalent à (P), c'est-à-dire que toute solution optimale de (P) soit solution optimale de (P').

Considérons pour cela le programme mathématique

$$(P') \quad \begin{cases} \text{Max } f(x) \\ a(x) \geqslant o \\ \varepsilon_i b_i(x) \geqslant o \quad , \; i = 1,\ldots q. \end{cases}$$

avec $\varepsilon_i = \pm 1$.

On passe de (P) à (P') en remplaçant les égalités par des inégalités. Le problème consiste à déterminer, si c'est possible, les ε_i de manière que (P') soit quasi-équivalent à (P).

Cette idée est basée sur le fait qu'une équation $b_i(x) = o$ peut être remplacée par deux inéquations

$$\begin{cases} b_i(x) \geqslant o \\ b_i(x) \leqslant o \end{cases}$$

et qu'à l'optimum de (P), une seule de ces deux contraintes sera active, bien que toutes les deux soient saturées ; en supprimant la contrainte non active, on retrouve la formulation de (P').

IV.2. Définition et propriété du programme (P').

On considère le programme mathématique

$$(P) \quad \left\| \begin{array}{l} \text{Maximiser } f(x) \\ \left.\begin{array}{l} a(x) \geqslant o \\ b(x) = o \end{array}\right\} A \end{array}\right.$$

où $x \in \mathbb{R}^n$, avec :

$a : \mathbb{R}^n \longrightarrow \mathbb{R}^p$, différentiable, de composantes a_i, $i = 1,2...p$

$b : \mathbb{R}^n \longrightarrow \mathbb{R}^q$, différentiable, de composantes b_j, $j = 1,2...q$

$f : \mathbb{R}^n \longrightarrow \mathbb{R}$, différentiable.

Le lagrangien de (P) est une fonction $L : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \longrightarrow \mathbb{R}$, définie par

$$L(x,v,w) = f(x) + v.a(x) + w.b(x)$$

En un point $\hat{x} \in A$, on pose

$$E = \{i \mid a_i(\hat{x}) = o\} \ , \ \bar{E} = \{i \mid a_i(\hat{x}) > o\}$$

$\hat{K}_1 = \{y \in \mathbb{R}^n \mid \nabla a_i(\hat{x}).y \geq o,\ i \in E,\ \nabla b_j(\hat{x}).y = o,\ j = 1,2,\ldots q\}$.

On fait l'hypothèse suivante (hypothèse du 1er ordre) :

Soient $T(A,\hat{x})$ le cône tangent en $\hat{x}$ à A et $P = \overline{[T(A,\hat{x})]}$ l'enveloppe convexe fermée de $T(A,\hat{x})$.

On suppose que $\hat{K}_1 = P$.

PROPOSITION IV.1 :

Si $\hat{x}$ est solution optimale de (P), il existe un programme mathématique (P'), ne comportant que des contraintes en inégalités, tel que $\hat{x}$ vérifie les conditions de Kuhn et Tucker de (P').

Démonstration :

Les conditions de Kuhn et Tucker sont nécessaires pour (P).
$\hat{x}$ solution optimale de (P) $\Longrightarrow$

$\exists v \in \mathbb{R}^p$, $w \in \mathbb{R}^q$, tels que

$$\begin{cases} v \geq o \\ \nabla L(\hat{x},v,w) = o \\ v.a(\hat{x}) = o \end{cases}$$

w est un vecteur ligne à composantes sans condition de signe.
Soit ε_j = signe (w_j) $\forall j = 1,\ldots q$, et considérons le vecteur
$u \in \mathbb{R}^q$ tel que $u_j = \varepsilon_j w_j$ $\forall j = 1,\ldots q$. (donc $u \geq o$).

$$\sum_{j=1}^{q} w_j b_j(\hat{x}) = \sum_{j=1}^{q} u_j \varepsilon_j b_j(\hat{x}).$$

Posons $c_j(x) = \varepsilon_j b_j(x)$ $\forall j = 1,\ldots q$

$$\Longrightarrow \sum_{j=1}^{q} w_j b_j(\hat{x}) = \sum_{j=1}^{q} u_j c_j(\hat{x})$$

Soit L' : $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \longrightarrow \mathbb{R}$, définie par

$L'(x,v,u) = f(x) + v.a(x) + u.c(x)$

On a $L(\hat{x},v,w) = L'(\hat{x},v,u)$.

D'autre part $b(\hat{x}) = o \Longrightarrow u.c(\hat{x}) = o$.

On peut donc écrire :

$\exists v \in \mathbb{R}^p$, $u \in \mathbb{R}^q$, tels que

$$\left|\begin{array}{l} v \geqslant o \\ u \geqslant o \\ a(\hat{x}) \geqslant o \\ c(\hat{x}) \geqslant o \\ \nabla L'(x,v,u) = o \\ v.a(\hat{x}) + u.c(\hat{x}) = o. \end{array}\right.$$

Considérons alors le programme mathématique

$$(P') \quad \left\|\begin{array}{l} \text{Maximiser } f(x) \\ a(x) \geqslant o \\ c(x) \geqslant o \end{array}\right. \quad \left.\begin{array}{l} \\ \\ \end{array}\right\} \; A'$$

$L'(x,v,u)$ est le lagrangien de (P').
Nous voyons donc que $\hat{x}$ vérifie les conditions de Kuhn et Tucker de (P').

IV.3 **Cas convexe**.

Si nous faisons l'hypothèse supplémentaire que la fonction f est concave et que le domaine A' de (P') est convexe, nous avons la propriété suivante :

PROPOSITION IV.2 :

Il existe un programme mathématique (P'), ne comportant que des contraintes en inégalités, quasi-équivalent à (P).

Démonstration :

D'après la proposition IV.1., si $\hat{x}$ est solution optimale de (P), $\hat{x}$ vérifie les conditions de Kuhn et Tucker de (P'). L'hypothèse supplémentaire précédente implique que $\hat{x}$ est solution optimale de (P').(P') est donc quasi-équivalent à (P).

Remarque :

La réciproque n'est pas nécessairement vraie. En effet, le fait que $\hat{x}$ vérifie les conditions de Kuhn et Tucker de (P') n'implique pas que $\hat{x}$ vérifie les conditions de Kuhn et Tucker de (P'). Donc il n'y a que quasi-équivalence et non équivalence.

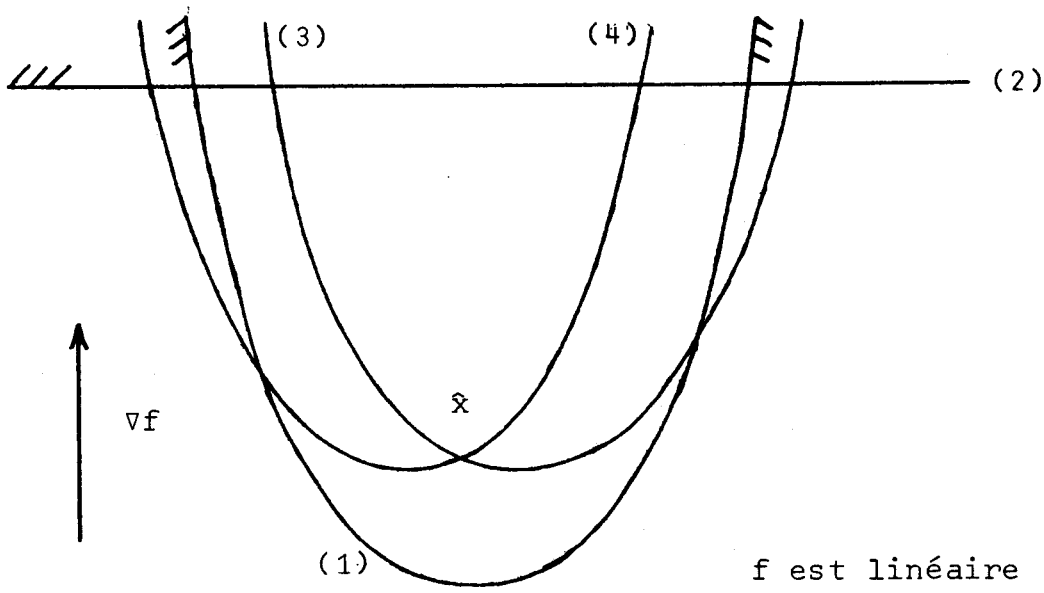
L'hypothèse de convexité précédente est vérifiée en particulier si f est concave, les fonctions a_i concaves et les fonctions b_j linéaires.

IV.4 Cas général.

IV.4.1 *Généralités :*

En général, le programme (P') n'est pas convexe, quelles que soient les hypothéses faites sur (P), et les conditions de Kuhn et Tucker ne sont pas dans ce cas des conditions suffisantes d'optimalité : la proposition IV.2 n'est donc pas vraie.

L'exemple suivant montre qu'il est possible qu'il n'existe pas de programme (P') quasi-équivalent au programme mathématique (P) initial :



f est linéaire
(1) et (2) en inégalité
(3) et (4) en égalité

Toutefois, on voit qu'il existe un programme (P') tel que $\hat{x}$ est optimum local de (P').

Nous allons montrer qu'il est possible de définir un programme (P') "localement quasi-équivalent" à (P), c'est-à-dire tel que toute solution optimale de (P) soit optimum local de (P'), moyennant certaines hypothèses, en particulier que les fonctions de (P) soient deux fois continuement différentiables.

IV.4.2 *Conditions suffisantes d'optimalité du second ordre.*

Nous rappelons d'abord un résultat de [18] et [26].

Considérons le programme mathématique

$$(M) \quad \left\| \begin{array}{l} \text{Maximiser } f(x) \\ x \in A \end{array} \right.$$

avec $A = \{x \in \mathbb{R}^n \mid g(x) \geq o,\ h(x) = o\}$.
$f : \mathbb{R}^n \longrightarrow \mathbb{R}$

$g : \mathbb{R}^n \longrightarrow \mathbb{R}^p$, de composantes g_i, $i = 1,\ldots p$
$h : \mathbb{R}^n \longrightarrow \mathbb{R}^q$, de composantes h_j, $j = 1,\ldots q$

Le lagrangien de (M) est une fonction

$L : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \longrightarrow \mathbb{R}$

définie par

$L(x,u,w) = f(x) + u.g(x) + w.h(x)$.

PROPOSITION IV.3 :

Si les fonctions du problème (M) sont deux fois continuement différentiables, et si $\exists \hat{x} \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $w \in \mathbb{R}^q$, tels que $(\hat{x},u,w)$ vérifie les relations

$$(R) \quad \left| \begin{array}{l} u \geqslant o \\ g(\hat{x}) \geqslant o \\ h(\hat{x}) = o \\ \nabla L(\hat{x},u,w) = o \\ u.g(\hat{x}) = o \end{array} \right.$$

et si $\forall y \in \mathbb{R}^n$, $y \neq o$, tel que $\nabla g_i(\hat{x}).y = o$ $\forall i \in \hat{D} = \{i \mid u_i > o\}$, $\nabla h_j(\hat{x}).y = o$, $j = 1,\ldots q$, et $\nabla f(\hat{x}).y = o$,

on a $y^t \nabla^2 L(\hat{x},u,w)\, y < o$,

alors $\hat{x}$ est un maximum local isolé de (M).

Démonstration :

Supposons que $\hat{x}$ n'est pas maximum local isolé de (M).
Il existe donc une suite de points $\{\overset{k}{x}\}$ telle que

$\lim_{k\to\infty} \overset{k}{x} = \hat{x}$, $\overset{k}{x} \in A$, et $f(\overset{k}{x}) \geq f(\hat{x})$ $\forall k$.

Posons $\overset{k}{x} = \hat{x} + \delta_k \overset{k}{y}$ $\forall k$, avec $||\overset{k}{y}|| = 1$, $\delta_k > 0$.

Il est clair que $\delta_k \to 0$ quand $k \to \infty$.

La boule $\{x \in \mathbb{R}^n \mid ||x|| \leq 1\}$ est compacte dans $\mathbb{R}^n$; la suite $\{\overset{k}{y}\}$ possède donc au moins un point d'accumulation ; soit y^* un tel point.

On a $||y^*|| = 1$.

$$\overset{k}{x} \in A \Longrightarrow \begin{array}{l} g_i(\overset{k}{x}) - g_i(\hat{x}) \geq 0 \ \forall i \in E = \{i \mid g_i(\hat{x}) = 0\}. \\ h_j(\overset{k}{x}) - h_j(\hat{x}) = 0 \ \forall j = 1,\ldots q. \end{array}$$

D'autre part $f(\overset{k}{x}) - f(\hat{x}) \geq 0$.

En divisant ces relations par δ_k et en faisant tendre k vers l'infini, on a à la limite

$$\begin{array}{l} \nabla g_i(\hat{x}).y^* \geq 0 \ \forall i \in E \\ \nabla h_j(\hat{x}).y^* = 0 \ j = 1,\ldots q \\ \nabla f(\hat{x}).y^* \geq 0 \end{array}$$

Par hypothése, $\nabla L(\hat{x}, u, w) = 0$

$$\Longrightarrow \nabla f(\hat{x}) = - \sum_{i=1}^{p} u_i \nabla g_i(\hat{x}) - \sum_{j=1}^{q} w_j \nabla h_j(\hat{x})$$

$$\Longrightarrow \underbrace{\nabla f(\hat{x}).y^*}_{\geq 0} = - \underbrace{\sum_{i=1}^{p} u_i \nabla g_i(\hat{x}).y^*}_{\geq 0} - \underbrace{\sum_{j=1}^{q} w_j \nabla h_j(\hat{x}).y^*}_{= 0}$$

Ce qui n'est possible que si

$\nabla g_i(\hat{x}).y^* = o \quad \forall i \in \hat{D}$

$\nabla f(\hat{x}).y^* = o$

Le développement de Taylor de $L(\overset{k}{x},u,w)$ donne

$$L(\overset{k}{x},u,w) = L(\hat{x},u,w) + \delta_k \overset{k}{y}{}^t \nabla L(\hat{x},u,w) + \frac{1}{2} \delta_k{}^2 \overset{k}{y}{}^t \nabla^2 L(\overset{k}{\eta},u,w) \overset{k}{y}$$

avec $\overset{k}{\eta} = \hat{x} + \alpha_k \delta_k \overset{k}{y}, \quad o < \alpha_k < 1.$

Donc $\frac{1}{2} \delta_k{}^2 \overset{k}{y}{}^t \nabla^2 L(\overset{k}{\eta},u,w) \overset{k}{y} =$

$$f(\overset{k}{x}) + \sum_{i=1}^{p} u_i g_i(\overset{k}{x}) + \underbrace{\sum_{j=1}^{q} w_j h_j(\overset{k}{x})}_{= o} - f(\hat{x}) - \underbrace{\sum_{i=1}^{p} u_i g_i(\hat{x})}_{= o} - \underbrace{\sum_{j=1}^{q} w_j h_j(\hat{x})}_{= o}$$

$$- \delta_k \overset{k}{y}{}^t \underbrace{\nabla L(\hat{x},u,w)}_{= o}$$

Ce terme est positif. En effet $u_i \geq o$, $g_i(\overset{k}{x}) \geq o \ \forall i$, et $f(\overset{k}{x}) \geq f(\hat{x})$.

$\Longrightarrow \overset{k}{y}{}^t \nabla^2 L(\hat{x} + \alpha_k \delta_k \overset{k}{y},u,w) \overset{k}{y} \geq o.$

En faisant tendre k vers l'infini, on obtient à la limite

$y^{*t} \nabla^2 L(\hat{x},u,w) y^* \geq o$

ce qui contredit l'hypothèse.

Remarque :

F I A C C O ([14]) affaiblit les hypothèses de la proposition précédente, de la façon suivante : on pose

$$y = \{y | \nabla g_i(\hat{x}).y = o \ \forall i \in \hat{D} \ ; \ \nabla h_j(\hat{x}).y = o, \ j = 1,..q \ ; ||y|| = 1\}.$$

$$z(\varepsilon,\delta) = \{z | \exists y \in Y, ||z - y|| \leq \varepsilon \ ; \ \exists \delta_z, \ o < \delta_z < \delta, \ \hat{x} + \delta_z z \in A \ ; ||z|| = 1\}$$

Si les fonctions du problème (M) sont deux fois continuement différentiables, et si $\exists u \in \mathbb{R}^p$, $w \in \mathbb{R}^q$, tels que $(\hat{x},u,w)$ vérifie les relations (R), et si $\exists \varepsilon > o$, $\delta > o$, tels que $\forall z \in Z(\varepsilon,\delta)$
on ait

$$z^t \nabla^2 L(\hat{x} + \lambda\delta_z z, u, w) \ z \leq o \quad \forall \lambda \text{ tel que } o < \lambda < 1,$$

alors $\hat{x}$ est un maximum local, non nécessairement isolé, de (M).

IV.4.3 *Existence du programme localement quasi-équivalent* :

Nous allons appliquer la proposition IV.3 à notre problème.

Soit $\hat{x}$ une solution optimale de (P). Les conditions de Kuhn et Tucker sont nécessaires pour (P) ; soient $v \in \mathbb{R}^p$ et $w \in \mathbb{R}^q$ les multiplicateurs.
On pose :

$$D_1 = \{i | v_i > o\}, \ D_2 = \{j | w_j \neq o\}$$

$$\hat{K}_2 = \{y \in \mathbb{R}^n | \nabla a_i(\hat{x}).y = o, i \in D_1, \nabla b_j(\hat{x}).y = o, j \in D_2, \nabla f(\hat{x}).y = o\}.$$

Nous faisons sur (P) l'hypothèse supplémentaire que les fonctions f, a et b sont deux fois continuement différentiables, ainsi que l'hypothèse suivante (hypothèse du 2e ordre) :

Soit $L = \{y \in \mathbb{R}^n \mid y^t \nabla^2 L(\hat{x},v,w)y < o\}$

On suppose que $\hat{K}_2 \subset L$.

PROPOSITION IV.4. :

Il existe un programme mathématique (P'), ne comportant que des contraintes en inégalités, localement quasi-équivalent à (P).

Démonstration ;

D'après la proposition IV.1, $\hat{x}$ vérifie les conditions de Kuhn et Tucker de (P').

$\exists v \in \mathbb{R}^p$, $u \in \mathbb{R}^q$, tels que

$$\left|\begin{array}{l} v \geqslant o \\ u \geqslant o \\ a(\hat{x}) \geqslant o \\ c(\hat{x}) \geqslant o \\ \nabla L'(\hat{x},v,u) = o \\ v.a(\hat{x}) + u.c(\hat{x}) = o \end{array}\right.$$

D'après la définition de (P') on peut écrire :

Si $\exists y$ tel que $\nabla b_j(\hat{x}).y = o$, alors $\nabla c_j(\hat{x}).y = o$.

D'autre part, $D_2 = \{j \mid u_j > o\}$ et $L(\hat{x},v,w) = L'(\hat{x},v,u)$.

Donc $\forall y \in \mathbb{R}^n$, $y \neq o$, tel que $\nabla a_i(\hat{x}).y = o$ $\forall i \in D_1$,

$\nabla c_j(\hat{x}).y = o$ $\forall j \in D_2$, et $\nabla f(\hat{x}).y = o$, on a

$y^t \nabla^2 L'(\hat{x},v,u)y < o$

Nous voyons que (P') vérifie les hypothèses de la proposition IV.3. Donc $\hat{x}$ est un maximum local isolé de (P').

Remarque 1 :

il est montré dans [26] que, si les vecteurs $\nabla a_i(\hat{x})$ $\forall i \in E$, et $\nabla b_j(\hat{x})$ $\forall j = 1,...q$, sont linéairement indépendants, on a

$y^t \nabla^2 L(\hat{x},v,w)y \leqslant o \quad \forall y \in \hat{K}_2$.

Toutefois, cette condition n'entraîne pas l'inégalité stricte.

Remarque 2 :

Il est possible d'affaiblir l'hypothèse du 2e ordre en utilisant la remarque de la proposition IV.3. Dans ce cas, $\hat{x}$ sera maximum local, mais non nécessairement isolé, de (P').

IV.5. Recherche itérative du programme (P').

Nous supposons qu'il existe un programme (P') quasi-équivalent à (P), et nous voulons le déterminer.

Si on ne connaît pas à priori le signe des multiplicateurs w_i, il y a 2^q programmes (P') possibles, et il n'est pas question de les résoudre tous. On préfère donc déterminer (P') en utilisant l'algorithme itératif décrit dans [12] : on remplace les contraintes en égalités par des inégalités de sens arbitraire, et on corrige ensuite ce choix jusqu'à ce que ces contraintes soient vérifiées en égalité à l'optimum du programme obtenu.

Considérons le programme mathématique (P). Nous supposerons f concave.

Etant donné un sous-ensemble d'indices S C {1,2,...q}, on pose :

$\bar{S} = \{1,2,...q\} - S$

$D(S) = \{x | b_i(x) \geq o, i \in S \; ; \; b_i(x) \leq o, \; i \in \bar{S} \; ; \; a(x) \geq o\}.$

et on définit le programme mathématique

$$P(S) \quad \left\| \begin{array}{l} \text{Maximiser } f(x) \\ x \in D(S) \end{array} \right.$$

Si P(S) admet une solution optimale $\hat{x}$ (S) telle que

$b(\hat{x}(S)) = o,$

alors $\hat{x}(S)$ est solution optimale de (P) et S représente le bon choix du sens des inéquations. Sinon :

$\exists s \in \{1,...q\}$ tel que $b_s(\hat{x}(S)) \neq o.$

On remplace alors S par S' tel que :

$S' = S + s$ si $s \notin S$
$\quad = S - s$ si $s \in S$

On modifie donc le sens de l'inégalité d'indice s.

Considérons $\hat{x}(S')$ optimum de P(S') et le segment $[\hat{x}(S'), \hat{x}(S)]$.

La contrainte $\{x | b_s(x) = o\}$ sépare les ensembles D(S) et D(S').
Or $\hat{x}(S)$ n'appartient pas à cette contrainte, $\hat{x}(S) \in D(S)$, et $\hat{x}(S') \in D(S')$; donc $[\hat{x}(S'), \hat{x}(S)]$ rencontre la contrainte en un point $y \neq \hat{x}(S)$.

PROPOSITION IV.5 :

Nous supposons que, quels que soient les ensembles S et S' définis comme précédemment, on a y $\in$ D(S).
Alors, ou bien $f(\hat{x}(S')) < f(\hat{x}(S))$, ou bien $f(\hat{x}(S')) = f(\hat{x}(S)) = f(y)$.

Démonstration :

$y \in D(S)$ et $\hat{x}(S)$ maximise f sur D(S) $\Longrightarrow f(y) \leqslant f(\hat{x}(S))$.

Supposons alors $f(\hat{x}(S')) > f(\hat{x}(S))$.

$y \in [\hat{x}(S'), \hat{x}(S)] \Longrightarrow y = \lambda\hat{x}(S') + (1 - \lambda)\hat{x}(S), \lambda \in [0,1]$.

$$f \text{ concave} \Longrightarrow f(y) \geqslant \lambda f(\hat{x}(S')) + (1 - \lambda) f(\hat{x}(S))$$
$$> \lambda f(\hat{x}(S)) + (1 - \lambda) f(\hat{x}(S))$$

Donc $f(y) > f(\hat{x}(S))$, ce qui est contradictoire.

D'où finalement : $f(\hat{x}(S')) \leqslant f(\hat{x}(S))$.

Si $f(\hat{x}(S')) = f(\hat{x}((S))$:

$$f(y) \geqslant \underbrace{\lambda f(\hat{x}(S')) + (1 - \lambda) f(\hat{x}(S))}_{= f(\hat{x}(S))}$$

Or $f(y) \leqslant f(\hat{x}(S))$; donc $f(y) = f(\hat{x}(S))$.

Remarque 1 :

L'hypothèse $y \in D(S)$ est vérifiée en particulier si les contraintes a sont concaves et les contraintes b linéaires.

Dans le cas général elle n'est pas nécessairement vérifiée ; cependant, dans la plupart des problèmes concrets, la non-linéarité des contraintes étant faible, on aura cette propriété.

Remarque 2 :

Si f, au lieu d'être concave, est strictement quasi-concave, on montre que dans ce cas on a toujours $f(\hat{x}(S')) < f(\hat{x}(S))$.

ALGORITHME DE RECHERCHE DE (P').

Itération k :

1) On dispose de $S_k \subset \{1,\ldots q\}$, $P(S_k)$
$\overset{k}{x}$ solution optimale de $P(S_k)$
Si $b_i(\overset{k}{x}) = o \ \forall i = 1,\ldots q$, alors $\overset{k}{x}$ est solution optimale de (P) : fin de l'algorithme.

Sinon, $\exists s \in \{1,\ldots q\} : b_s(\overset{k}{x}) \neq o$.

Soit $S'_k = S_k + s$ si $s \notin S_k$
$= S_k - s$ si $s \in S_k$

2) Calculer $\overset{k}{x}'$ solution optimale de $P(S'_k)$.

Si $f(\overset{k}{x}') < f(\overset{k}{x})$, prendre $S_{k+1} = S'_k$, $\overset{k+1}{x} = \overset{k}{x}'$, et aller en 1) avec $k = k + 1$.

Si $f(\overset{k}{x}') = f(\overset{k}{x})$, déterminer le point $\overset{k}{y}$ intersection du segment $[\overset{k}{x}, \overset{k}{x}']$ avec la contrainte

$$\{x \mid b_s(x) = o\} ;$$

modifier $\overset{k}{x}$ en prenant $\overset{k}{x} = \overset{k}{y}$, laisser S_k inchangé et aller en 1).

PROPOSITION IV.6 : (convergence).

Si f est concave ou strictement quasi-concave, si nous avons l'hypothèse précédente y € D(S) ∀S et S', et si de plus chaque itération k fournit un nombre fini de points $\overset{k}{x}'$, l'algorithme converge en un nombre fini d'itérations, c'est-à-dire que ∃k* € ℕ : $\overset{k^*}{x}$ solution optimale de (P).

Démonstration :

On obtient une suite de points $\overset{k}{x}$, chacun d'eux en un nombre fini d'étapes, telle que $f(\overset{k+1}{x}) < f(\overset{k}{x})$ ∀k. Donc on ne peut retrouver un ensemble S_k déjà rencontré, et comme les S_k sont en nombre fini (2^q), on arrivera nécessairement à la bonne répartition des sens au bout d'un nombre fini d'itérations.

Remarque 1 :

Si f est strictement quasi-concave, on a toujours $f(\overset{k}{x}') < f(\overset{k}{x})$; chaque itération k fournit donc un point $\overset{k}{x}'$ unique, ce qui implique la convergence
Si f est concave, il peut, en toute rigueur, se produire un cyclage au cours d'une itération k, bien que le fait de modifier $\overset{k}{x}$ en prenant $\overset{k}{x} = \overset{k}{y}$ dans le cas où $f(\overset{k}{x}') = f(\overset{k}{x})$ empêche le cyclage immédiat.

Remarque 2 :

Si l'on résout $P(S_k)$ par une méthode intérieure, il n'est pas nécessaire d'optimiser complètement le problème ; en effet, les indices s possibles sont en général connus au bout d'un petit nombre d'itérations de la méthode.

Remarque 3 :

Dans de nombreux problèmes concrets, il est possible de déterminer a priori le sens des inéquations, grâce à des considérations d'ordre économique. Nous reviendrons sur cette idée en V.3.1.

IV.6. Expériences numériques.

L'algorithme de IV.5. a été appliqué à deux des problèmes cités en annexe, avec la méthode des centres linéarisée. Comme nous l'avons remarqué, il est inutile d'optimiser complètement les programmes $P(S_k)$; quelques itérations de la méthode suffisent pour déterminer si les valeurs des contraintes se rapprochent de o ou non.

1er essai : Problème n°3 (voir annexe A1).

Après élimination des variables P_i et Q_i, ce problème comporte 2 contraintes en équation. Il y a donc 4 répartitions possibles des sens des inéquations. Nous avons résolu chacun de ces 4 cas .

Valeurs des contraintes en fonction du numéro de troncature :

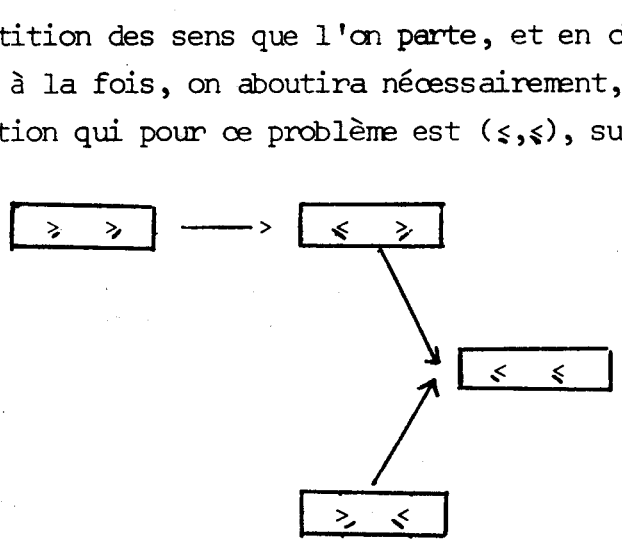
Troncature	contrainte 1 $\geq$	contrainte 2 $\geq$
1	1695,35	25,79
2	1795,53	0,84
3	1800,69	0,30
4	1801,85	0,18
5	1806,58	0,09
6	1810,62	0,03
7	1810,96	0,00
8	1810,96	

Troncature	contrainte 1 $\geq$	contrainte 2 $\leq$
1	1709,22	14,65
2	1789,04	58,33
3	1806,16	0,25
4	1801,90	0,22
5	1805,78	0,04
6	1807,41	0,02
7	1808,53	0,00
8	1809,60	

Troncature	contrainte 1 ≤	contrainte 2 ≥
1	8,91	140,51
2	0,22	225,21
3	0,02	245,01
4	0,00	251,49
5		254,34
6		255,82
7		257,19
8		258,29

Troncature	contrainte 1 ≤	contrainte 2 ≤
1	7,41	6,24
2	3,50	2,76
3	2,31	1,79
4	1,91	1,46
5	1,51	1,15
6	1,36	1,03
7	1,16	0,87
8	1,05	0,78

La décroissance ou non-décroissance des fonctions est nette et on voit que, de quelque répartition des sens que l'on parte, et en changeant de sens une seule contrainte à la fois, on aboutira nécessairement, dans tous les cas, à la bonne répartition qui pour ce problème est $(\lesssim,\lesssim)$, suivant le schéma ci-dessous :



2e essai : problème n°6 (voir annexe A1)

Ce problème comporte 88 contraintes d'égalités et il y a donc 2^{88} répartitions possibles des sens des inégalités. Cependant, comme on le verra en V.3.1, les contraintes du réactif correspondant aux noeuds producteurs 11,24,25,29,44 ne sont pas nécessairement actives à l'optimum (ce sont les contraintes 56,69,70,74,89). Les 83 autres contraintes doivent être actives.

Au départ, toutes les égalités sont transformées en inégalités en $\leq$. On laisse se dérouler le code sur quelques itérations et on change de sens les contraintes dont l'évolution n'est pas satisfaisante : et ainsi de suite jusqu'à ce qu'on ait obtenu la bonne répartition des sens.

1e étape :

Toutes les contraintes sont en $\leq$. Toutes les contraintes actives se rapprochent de o, sauf :

tronçature \ contrainte	51	52	53	63	73
1	0,5665	0,6776	1,2401	0,1768	0,6112
2	5,9492	0,4885	0,8938	15,084	5,8334
3	1,5382	4,0410	2,3903	14,422	8,6847
4	0,8688	2,8061	1,1892	6,6440	3,9037
5	4,3672	0,6762	1,2888	5,2408	7,5958
6	0,6723	4,9700	1,7317	7,8478	8,7955
7	4,8303	0,7090	2,1620	8,9323	9,0560
8	2,7441	2,8131	2,5963	10,929	9,2878
9	0,6216	4,9513	2,8855	12,358	9,4401

2e étape :

Les contraintes 51,52,53,63,73, sont en $\geq$ et toutes les autres en $\leq$. Toutes les contraintes actives se rapprochent de o, sauf :

troncature \ contrainte	55	62	64	67
1	1,9417	1,9074	19,475	25,499
2	1,0115	16,126	9,2963	29,066
3	0,4214	5,8193	6,6224	27,171
4	2,3474	1,9979	2,0783	26,482
5	0,6452	0,6854	10,858	26,784
6	5,7905	0,2739	2,9237	26,948
7	0,9874	0,1036	12,147	26,807
8	5,8532	0,0541	3,1709	26,889
9	3,2914	1,1532	7,1059	26,977

3e étape :

Les contraintes 51,52,53,55,62,63,64,67,73, sont en $\geq$ et toutes les autres en $\leq$. Toutes les contraintes actives se rapprochent de 0, sauf :

troncature \ contrainte	54	66
1	0,0781	52,703
2	21,292	27,165
3	7,3466	38,590
4	12,362	12,397
5	3,6763	22,521
6	5,7053	21,418
7	0,9092	28,638
8	1,4921	30,145

4é étape :

Les contraintes 51,52,53,54,55,62,63,64,66,67,73, sont en $\geq$ et toutes les autres en $\leq$. Toutes les contraintes actives se rapprochent de 0, sauf :

contrainte / troncature	61	65
1	13,624	58,047
2	41,125	27,153
3	23,324	32,642
4	7,9337	40,085
5	25,156	13,677
6	9,1119	17,321
7	1,9830	27,005
8	5,8723	29,023
9	11,137	20,757

5e étape :

Les contraintes 51,52,53,54,55,61,62,63,64,65,66,67,73, sont en $\geq$ et toutes les autres en $\leq$. Toutes les contraintes actives se rapprochent de o, sauf :

contrainte / capture	60	68
1	38,557	1,0680
2	20,965	0,8149
3	11,751	43,717
4	35,009	13,829
5	14,306	5,7598
6	23,426	1,2681
7	16,656	12,774
8	22,932	1,8071
9	17,938	9,2397

6e étape :

Les contraintes 51,52,53,54,55,60,61,62,63,64,65,66,67,68,73 sont en $\geq$ et toutes les autres en $\leq$.

Toutes les contraintes actives se rapprochent de 0.

On a donc déterminé le sens correct des inégalités. Les contraintes dont le sens est $\geq$ sont les contraintes du réactif correspondant aux noeuds 6,7,8,9,10,15,16,17,18,19,20,21,22,23,28.

On voit que pour obtenir ce résultat, il a suffi de résoudre (partiellement) 6 problèmes, au lieu de 2^{83}.

CHAPITRE V

LE DISPATCHING ECONOMIQUE

V.1 Introduction :

Nous avons expérimenté la méthode des centres linéarisée, ainsi que les diverses variantes exposées dans les chapitres précédents, sur un problème d'origine concrète : le Dispatching Economique.

Ce problème consiste, étant donné un réseau de production et de transport d'énergie électrique, à déterminer les puissances des usines génératrices en service, ainsi que les tensions et déphasages des courants aux divers noeuds, de façon à satisfaire les consommations et à minimiser les frais de production.

Le Dispatching Economique a été très étudié et abordé par différentes méthodes ; on trouvera dans [13] une approche réalisée à l'aide de modèles linéaires, ainsi que la bibliographie concernant le problème.

Nous considérons ici le modèle non linéaire obtenu par application de la loi des mailles, tel qu'il est formulé dans [6].

Après avoir exposé les modifications du modèle nécessitées par l'utilisation de la méthode, nous donnons les techniques de calcul utilisées pour la résolution du problème, compte tenu de ses particularités en structure et en valeurs numériques.

V.2 Formulation du problème :

V.2.1 *Notations.*

Dans ce chapitre, nous respectons les notations utilisées couramment dans la littérature spécifique d'application consacrée au Dispatching Economique, dans la mesure où cela n'introduit pas d'ambiguité avec les notations générales.

On considère un réseau de transport à n + 1 noeuds 0,1,2...n.

Au noeud i, on appelle :

P_i la production de puissance active

C_i la consommation de puissance active

$I_i = P_i - C_i$ l'injection de puissance active dans le réseau

Q_i la production de puissance réactive

D_i la consommation de puissance réactive

$K_i = Q_i - D_i$ l'injection de puissance réactive dans le réseau

U_i le module de la tension

θ_i la phase de la tension

Soit e(i) l'ensemble des noeuds voisins de i

Pour $j \in e(i)$, soient A_{ij}, B_{ij}, α_{ij}, β_{ij}, des constantes de la branche ij.

V.2.2 *Modèle mathématique:*

Avec les notations précédentes, l'application de la loi des noeuds de Kirchhoff permet d'exprimer les injections I_i et K_i de puissance active et réactive au noeud i :

$$I_i(\theta,U) = - \sum_{j\in e(i)} \frac{U_i U_j}{B_{ij}} \cos(\theta_i - \theta_j + \beta_{ij}) + \sum_{j\in e(i)} U_i^2 \frac{A_{ij}}{B_{ij}} \cos(\beta_{ij} - \alpha_{ij})$$

$$K_i(\theta,U) = - \sum_{j\in e(i)} \frac{U_i U_j}{B_{ij}} \sin(\theta_i - \theta_j + \beta_{ij}) + \sum_{j\in e(i)} U_i^2 \frac{A_{ij}}{B_{ij}} \sin(\beta_{ij} - \alpha_{ij})$$

On veut déterminer les puissances actives P_i et réactives Q_i à produire aux usines génératrices, les phases θ_i et les modules U_i des tensions en tous les noeuds de façon à minimiser les frais de production $F(P_0, P_1, \ldots, P_n)$.
Les demandes C_i et D_i sont fixées en tout noeud.
Les variables P_i, Q_i, U_i, θ_i sont soumises aux conditions suivantes :

a) <u>relations d'injection</u> :

$$I_i(\theta, U) - P_i + C_i = 0$$
$$K_i(\theta, U) - Q_i + D_i = 0$$

b) <u>Inéquations exprimant les limitations sur les productions et les tensions</u> :

$$P_i^2 + Q_i^2 - S_i^2 \leqslant 0$$

$$P_i^m \leqslant P_i \leqslant P_i^M$$

$$Q_i^m \leqslant Q_i \leqslant Q_i^M$$

$$U_i^m \leqslant U_i \leqslant U_i^M$$

(S_i, $P_i^m, U_i^m, P_i^M, Q_i^M, U_i^M$ étant des constantes).

c) <u>Inéquations exprimant les limitations de transits sur les lignes</u> :

$$\theta_i - \theta_j - T_{ij} \leqslant 0 \qquad (T_{ij} \text{ constante})$$

pour tout couple ordonné de noeuds voisins ij.

Les variables θ_i, n'intervenant que par des différences, sont définies à une constante près. On s'impose donc $\theta_0 = 0$ (le noeud 0 est noeud de référence).

V.2.3 *Remarques pratiques* :

1) Les contraintes du problème

Le problème du Dispatching Economique ainsi défini se présente sous la forme d'un programme mathématique comportant des contraintes de bornes sur les variables P_i, Q_i et U_i, des contraintes linéaires en inéquation, des contraintes non linéaires en inéquation et des contraintes non linéaires en équation.

Mais dans la pratique, il arrive fréquemment que certaines de ces contraintes n'apparaissent pas :

a) Certains problèmes sont actifs et non réactifs. Ils ne comportent donc pas de variables Q_i, et les relations d'injection relatives aux puissances réactives n'existent pas.

b) Certains problèmes ne comportent pas de limitations de transits. La partie linéaire du problème se limite alors aux bornes sur les variables.

c) Les inéquations de la forme $P_i^2 + Q_i^2 - S_i^2 \leq 0$ apparaissent rarement. Ceci est sans doute dû au fait que, lorsqu'elles apparaissent, elles ne sont pas actives et l'on peut donc les supprimer.

2) La fonction économique

Nous devons minimiser $F(P_0, P_1, \ldots, P_n)$, la fonction F représentant les frais de production et ne dépendant que des puissances actives P_i. Cette fonction peut avoir des formes diverses ; cependant un cas particulier très fréquent est celui où elle est linéaire par morceaux.

3) Ordre de grandeur des données

Les valeurs des A_{ij} sont voisines de 1. Les valeurs de $\frac{1}{B_{ij}}$ sont en général comprises entre 10^{-2} et 10^{-1}. Les α_{ij} ont des valeurs faibles, comprises entre 10^{-3} et 10^{-2} en général. Les β_{ij} prennent leurs valeurs entre 1 et 1,5. Les C_i et D_i peuvent aller de 0 à quelques centaines d'unités.

V.2.4 *Exemples de réseaux*

Nous donnons en annexe (A1) quelques exemples de réseaux ayant servi pour les expériences numériques. Les quatre premiers sont des réseaux-tests, de petite taille (de 2 à 9 variables). Le cinquième est un réseau-test de plus grande taille (50 variables). Le problème n°6 est basé sur un réseau réel, le réseau grec, et possède une taille importante (103 variables, 88 contraintes non linéaires, et des contraintes de bornes sur les variables).

V.3 Adaptation du modèle :

V.3.1 *Les contraintes en équation*

Le problème du Dispatching Economique tel que nous l'avons énoncé ne peut être résolu par la Méthode des Centres. En effet, celle-ci impose que l'intérieur du domaine A (voir I.2.2) soit non vide, ce qui interdit les contraintes non linéaires en équation.

Nous allons donc chercher un modèle équivalent au modèle initial, mais qui ne comporte plus de contraintes non linéaires en équation.

1) Réduction du nombre de contraintes en équation

Considérons la contrainte

$$I_i(\theta,U) - P_i + C_i = 0.$$

Nous remarquons que dans l'expression $I_i(\theta,U)$, la variable P_i n'intervient pas ; d'autre part, cette variable n'intervient que dans une seule contrainte de la forme ci-dessus; la variable P_i s'exprime donc en fonction des θ_i et des U_i :

$$P_i = I_i(\theta,U) + C_i.$$

De même, de la contrainte

$$K_i(\theta,U) - Q_i + D_i = 0$$

nous déduisons que la variable Q_i s'exprime en fonction des θ_i et des U_i :

$$Q_i = K_i(\theta,U) + D_i.$$

Nous pouvons donc supprimer du problème les variables P_i et Q_i en les remplaçant par leur expression en fonction des θ_i et des U_i.

La fonction économique $F(P_Q,P_1,\ldots P_n)$ devient une fonction $G(\theta,U)$.

D'où le nouvel énoncé du problème :

$$\begin{cases} \text{Minimiser } G(\theta,U) \\ \text{sous les contraintes} \\ [I_i(\theta,U) + C_i]^2 + [K_i(\theta,U) + D_i]^2 - S_i^2 \leqslant 0 \\ P_i^m \leqslant I_i(\theta,U) + C_i \leqslant P_i^M \end{cases}$$

$$\begin{cases} Q_i^m \leqslant K_i(\theta,U) + D_i \leqslant Q_i^M \\ U_i^m \leqslant U_i \leqslant U_i^M \\ \theta_i - \theta_j - T_{ij} \leqslant 0 \qquad j \in e(i) \end{cases}$$

Remarque 1 :

Par rapport au modèle initial, le nombre de variables est diminué de 2 fois le nombre de noeuds du réseau. Par contre, les contraintes en équation sont remplacées par des doubles contraintes en inéquation ; ce procédé a donc l'inconvénient d'augmenter, en définitive, le nombre total des contraintes du problème.

Remarque 2 :

Si l'on se fixe :

$$P_{i_o}^m = P_{i_o}^M = 0,\ P_{i_1}^m = P_{i_1}^M = 0,\ldots$$

$$Q_{j_o}^m = Q_{j_o}^M = 0,\ Q_{j_1}^m = Q_{j_1}^M = 0,\ldots$$

on retrouve dans le problème des contraintes en équation :

$$I_{i_o}(\theta,U) + C_{i_o} = 0,\ I_{i_1}(\theta,U) + C_{i_1} = 0,\ldots$$

$$K_{j_o}(\theta,U) + D_{j_o} = 0,\ K_{j_1}(\theta,U) + D_{j_1} = 0,\ldots$$

Ceci se produit fréquemment dans la pratique : c'est le cas particulier des noeuds non producteurs. On est dès lors ramené aux inconvénients du modèle initial.

2) Transformation des contraintes en équation en contraintes en inéquation :

En utilisant les remarques de IV.1, nous remplaçons les équations par des inéquations.

Compte tenu du fait que la fonction économique à minimiser est un coût de production, nous pouvons, dans une certaine mesure, prévoir le sens à donner aux inéquations.

Considérons tout d'abord la relation d'injection de puissance active

$$I_i(\theta,U) + C_i = P_i.$$

Si nous la mettons sous la forme

$$I_i(\theta,U) + C_i \leqslant P_i$$

cela signifie que la puissance disponible au noeud i, c'est-à-dire $P_i - C_i$, peut être supérieure à l'injection de puissance dans le réseau. Mais comme les frais de production sont d'autant moins élevés que cette quantité est plus faible, il est logique de penser qu'à l'optimum du problème on aura en général l'égalité.

En toute rigueur ce raisonnement est faux ; il est possible de trouver des exemples où le sens correct serait $\geqslant$. Mais dans le pratique, les problèmes traités vérifiaient le sens indiqué.

En ce qui concerne les relations d'injection réactive, le problème est différent. Ces relations sont mises sous la forme

$$K_i(\theta,U) + D_i - Q_i \leqslant 0.$$

Les variables Q_i interviennent linéairement dans ces contraintes et n'interviennent pas dans la fonction économique. Elles se comportent donc comme des variables d'écart. Il s'ensuit qu'en général à l'optimum du problème on trouvera des valeurs $\hat{Q}_i$ telles que

$$K_i(\theta,U) + D_i - \hat{Q}_i < 0.$$

On obtiendra alors une solution réalisable optimale en remplaçant $\hat{Q}_i$ par $K_i(\hat{\theta},\hat{U}) + D_i$.

Ceci n'est possible que si la nouvelle valeur de Q_i est comprise entre ses bornes. Dans le cas contraire, le sens correct de l'inéquation est $\geq$, ce qui peut se produire en pratique (voir le problème n°6 de l'annexe A1).

Ces remarques nous conduisent à utiliser l'algorithme défini en IV.5, de la manière suivante :

On remplace les contraintes en égalités par des inégalités en $\leq$. Si, comme c'est souvent le cas, le choix des sens est bon, les contraintes ainsi modifiées seront actives à l'optimum. Sinon, on changera de sens les contraintes dont l'évolution n'est pas satisfaisante, et ainsi de suite.

Nous avons vu en IV.6 que cet algorithme, appliqué au problème n°6 de l'annexe A1, converge très rapidement.

Sur les problèmes 1 à 5, on constate que toutes les inégalités doivent être en $\leq$ et que l'algorithme est inutile.

V.3.2 *Problèmes à fonction économique non continuement différentiable* :

Un cas particulier très fréquent du problème du Dispatching Economique est celui où la fonction économique est linéaire par morceaux : c'est le cas des problèmes n° 4, 5 et 6 (voir annexe A1). Le gradient de la fonction économique est alors discontinu, et une des hypothèses de la Méthode des Centres n'est pas vérifiée. Dans la pratique, il se produit sur la valeur des variables des oscillations très fortes qui ralentissement considérablement la convergence et peuvent même l'empêcher totalement (voir VI).

Rappelons tout d'abord comment sont obtenus ces problèmes à fonction économique non continuement différentiable. Le problème d'origine du Dispatching Economique est de la forme

$$\text{Minimiser} \sum_i \sum_{j \in J_i} \lambda_{ij} P_{ij}$$

sous les contraintes

$$I_i + C_i - \sum_{j \in J_i} P_{ij} \leq 0$$

$$K_i + D_i - Q_i \leq 0$$

$$x \in B$$

Pour un noeud i, les indices $j \in J_i$ représentent les différentes usines implantées en i.

Les variables sont les U_i, les θ_i, les P_{ij} et les Q_j.

Les fonction économique est linéaire.

D'autre part, on a :

$$\forall i,\ \lambda_{ij_2} > \lambda_{ij_1} \quad \text{si } j_2 > j_1.$$

Cette remarque permet d'effectuer le changement de variables

$$P_i = \sum_{j \in J_i} P_{ij},$$

ce qui entraîne une diminution du nombre total des variables. La fonction économique devient dès lors une fonction des P_i par l'intermédiaire des P_{ij}, qui sont eux-mêmes calculés à partir des variables P_i de la manière suivante :

Soit $P_{ij}^m \leq P_{ij} \leq P_{ij}^M$, P_{ij}^m et P_{ij}^M étant les bornes sur les P_{ij}

d'où $P_i^m = \sum_{j \in J_i} P_{ij}^m$ et $P_i^M = \sum_{j \in J_i} P_{ij}^M$ les bornes sur les P_i).

(1) Soit $P_i^{(0)} = P_i$. Faire $j = 1$.

(2) Si $P_i^{(j-1)} \leq P_{ij}^M$, faire $P_{ij} = P_i^{(j-1)}$ et $P_{ik} = 0 \quad \forall k > j, k \in J_i$

et aller en (4).

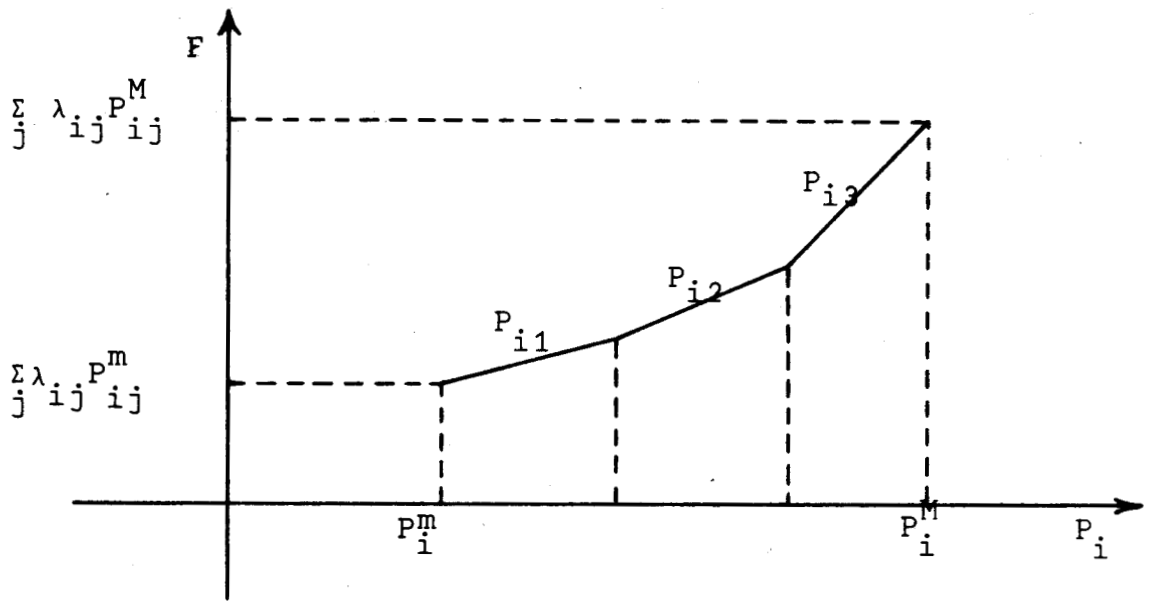
Sinon, faire $P_{ij} = P_{ij}^M$ et $P_i^{(j)} = P_i^{(j-1)} - P_{ij}^M$,

et aller en (3).

(3) Faire $j = j + 1$
et aller en (2).

(4) Fin.

La fonction économique est donc une fonction linéaire par morceaux des variables P_i.



Conclusion :

Le problème d'origine possède une fonction économique totalement linéaire. En conservant la forme d'origine nous n'avons donc plus l'inconvénient d'un gradient discontinu. Cette forme est donc préférable, même si le nombre des variables du problème s'en trouve augmenté.

V.3.3 *Convexité du problème* :

Considérons, pour une valeur donnée de la troncature, le tronçon $E(\lambda)$ défini par les contraintes

$$\begin{cases} f(x) - \lambda \geq 0 \\ g_i(x) \geq 0 \quad i \in L. \end{cases}$$

Lorsque les premiers membres sont concaves, le domaine est convexe, ce qui permet de démontrer la convergence théorique de la méthode. Dans la pratique, il n'en est pas toujours ainsi ; nous nous intéresserons en fait à des propriétés de "concavité locale" des fonctions, en particulier au voisinage de l'optimum.

Considérons d'abord le problème simplifié où les fonctions sont soit concaves, soit convexes dans le domaine de variation de x.

Soit y un point de linéarisation. Le domaine linéarisé est défini par les contraintes linéaires

$$\begin{cases} f'(x;y) - \lambda \geq 0 \\ g_i'(x;y) \geq 0 \quad i \in L. \end{cases}$$

avec

$$f'(x;y) = f(y) + \nabla f(y).(x-y)$$

$$g_i'(x;y) = g_i(y) + \nabla g_i(y).(x-y) \quad i \in L.$$

Considérons une fonction g_{i_o} quelconque et sa linéarisation g'_{i_o}.

Si g_{i_o} est concave nous avons

$$\nabla g_{i_o}(y).(x-y) \geq g_{i_o}(x) - g_{i_o}(y)$$

$$\Longrightarrow g'_{i_o}(x;y) \geq g_{i_o}(x).$$

Si g_{i_o} est convexe nous avons de même

$$g'_{i_o}(x;y) \leq g_{i_o}(x).$$

1) Soit x_o un point de l'hyperplan de linéarisation :

$$\Longrightarrow g'_{i_o}(x_o;y) = 0.$$

D'après les inégalités précédentes :

$$g_{i_o} \text{ concave} \Longrightarrow g_{i_o}(x_o) \leq 0$$

$$g_{i_o} \text{ convexe} \Longrightarrow g_{i_o}(x_o) \geq 0.$$

2) Soit x_1 un point du tronçon $E(\lambda)$:

$$\Longrightarrow g_{i_o}(x_1) \geq 0.$$

D'après les inégalités précédentes :

$$g_{i_o} \text{ concave} \Longrightarrow g'_{i_o}(x_1;y) \geq 0.$$

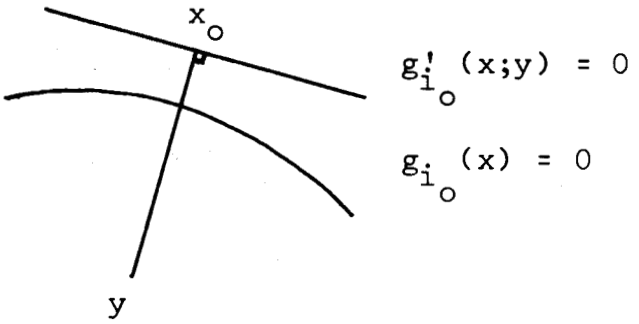
Le raisonnement précédent est valable pour la fonction f à condition de remplacer $g_{i_o}(x)$ par $f(x) - \lambda$.

Dans la pratique, les fonctions considérées sont quelconques et le raisonnement précédent se traduira par des propriétés de "concavité locale", à savoir la position de l'hyperplan de linéarisation par rapport à la contrainte.

Au cours d'expériences numériques sur des problèmes d'essais, il peut être utile d'effectuer des tests à titre d'analyse, pour étudier la forme des domaines.

1) <u>1er test de "convexité"</u> :

On projette le point de linéarisation y sur l'hyperplan de linéarisation ; soit x_o la projection. On détermine alors en x_o le signe de la contrainte non linéaire.



Détermination du point x_o :

Il est tel que

$$\begin{cases} x_o - y = \alpha[\nabla g_{i_o}(y)]^t \\ g_{i_o}(y) + \nabla g_{i_o}(y).(x_o - y) = 0 \end{cases}$$

$$\Longrightarrow \quad g_{i_o}(y) + \alpha[\nabla g_{i_o}(y)]^2 = 0 \Longrightarrow \alpha = -\frac{g_{i_o}(y)}{[\nabla g_{i_o}(y)]^2}$$

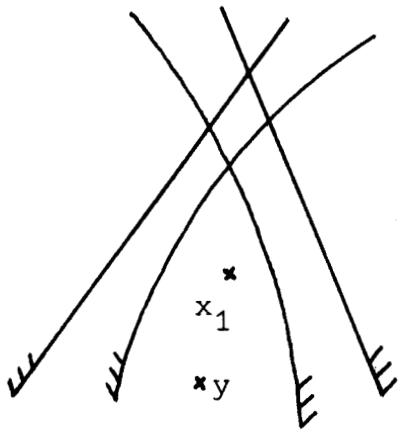
$$\text{Donc } x_o = y - \frac{g_{i_o}(y)\,[\nabla g_{i_o}(y)]^t}{[\nabla g_{i_o}(y)]^2}.$$

Pour la fonction f on a de même

$$x_o = y - \frac{[f(y) - \lambda]\,[\nabla f(y)]^t}{[\nabla f(y)]^2}$$

2) 2ème test de "convexité"

Soit un point réalisable x_1 de $E(\lambda)$. On calcule en ce point les signes des contraintes linéarisées $f'(x;y) - \lambda$ et $g'_i(x;y)$. Si le polyèdre linéarisé contient $E(\lambda)$, toutes ces fonctions doivent être positives.



3) Exemple numérique

Considérons le problème n°5 (voir annexe A1).

1er cas :

les variables P_i et Q_i ont été éliminées ; le problème comporte donc des doubles inéquations (voir V.3.1)

Les 2 tests de "convexité" ont été effectués, en un point y de linéarisation correspondant à une valeur de fonction économique à environ 5 unités de l'optimum.

Test n°1 :

valeurs de la fonction économique et des contraintes aux points x_o projections de y sur les hyperplans de linéarisation correspondants :

3.10^{-7}	1.10^{-2}	-2	8.10^{-2}	-1	9.10^{-1}
-2.10^{-1}	4.10^{-5}	-1	1	-1	3.10^{-1}
-1.10^{-1}	4.10^{-1}	-2.10^{-1}	1.10^{2}	-1.10^{2}	2.10^{1}
-2.10^{2}	2.10^{2}	-2.10^{-5}	7.10^{1}	-3.10^{1}	1.10^{2}
-4.10^{1}	1.10^{2}	-1.10^{2}	9.10^{1}	-9.10^{1}	-4.10^{-5}
-3.10^{-5}	-2.10^{-5}	-6.10^{-5}	-8.10^{-5}	-4.10^{-7}	

Les signes de ces valeurs donnent la position des hyperplans de linéarisation par rapport aux contraintes.

Test n°2 :

valeurs de la fonction économique et des hyperplans de linéarisation en x_1 optimum du problème :

6	4.10^{1}	4.10^{2}	6.10^{1}	3.10^{2}	3.10^{2}
2.10^{2}	6.10^{-2}	2.10^{2}	2.10^{2}	2.10^{2}	8.10^{1}
5.10^{1}	2.10^{2}	1.10^{2}	2.10^{2}	2.10^{2}	6.10^{1}
2.10^{2}	5.10^{2}	1.10^{-1}	2.10^{2}	1.10^{2}	2.10^{2}
1.10^{2}	8.10^{1}	6.10^{1}	1.10^{2}	1.10^{2}	-4.10^{-1}
1.10^{-1}	1.10^{-2}	8.10^{-3}	-5.10^{-2}	4.10^{-2}	

Les deux valeurs négatives prouvent que le polyèdre linéarisé ne contient pas le tronçon $E(\lambda)$.

2ème cas :

les variables P_i et Q_i n'ont pas été éliminées (voir V.3.1). Le test n°1 a été effectué en un point y de linéarisation correspondant à une valeur de fonction économique à environ 10^{-4} de l'optimum.

Valeurs de la fonction économique et des contraintes aux points x_o projections de y sur les hyperplans de linéarisation correspondants :

-4.10^{-6}	-1.10^{-7}	1.10^{-8}	2.10^{-8}	-1.10^{-7}	-6.10^{-9}
-3.10^{-8}	-4.10^{-8}	1.10^{-8}	-4.10^{-8}	0	-8.10^{-5}
-5.10^{1}	1.10^{-8}	0	1.10^{-8}	-6	1.10^{-9}
-1.10^{1}	-3.10^{-1}	-7.10^{1}			

Conclusion :

Dans le 1er cas nous voyons que le problème est nettement "non-convexe"

Dans le 2ème cas par contre l'examen des valeurs numériques montre que la "non-convexité" est faible. En pratique, bien que pour des fonctions non concaves la convergence de la méthode ne soit pas assurée, cette "non-convexité" n'empêche pas d'arriver à l'optimum (voir VI).

V.4. Organisation pratique des calculs.

V.4.1 *Maximisation de la F-distance linéarisée*.

Nous utilisons la méthode simpliciale avec calculs directs, telle que nous l'avons exposéeen II.2.7. Ceci permet le stockage de la matrice A sous forme "condensée", ce qui donne un gain de place mémoire si la matrice est très creuse.

Examinons donc la forme de la matrice A associée au problème du Dispatching Economique.

En chaque noeud i du réseau nous avons des contraintes de la forme

$$- \sum_{j \in e(i)} \frac{U_i U_j}{B_{ij}} \cos(\theta_i - \theta_j + \beta_{ij}) + \sum_{j \in e(i)} U_i^2 \frac{A_{ij}}{B_{ij}} \cos(\beta_{ij} - \alpha_{ij}) + C_i \leq P_i$$

$$- \sum_{j \in e(i)} \frac{U_i U_j}{B_{ij}} \sin(\theta_i - \theta_j + \beta_{ij}) + \sum_{j \in e(i)} U_i^2 \frac{A_{ij}}{B_{ij}} \sin(\beta_{ij} - \alpha_{ij}) + D_i \leq Q_i.$$

A chaque itération, la matrice du programme linéaire contient les dérivées partielles de ces contraintes par rapport aux variables : les U_i, les θ_i, les P_i (ou les P_{ij}), et les Q_i. Donc dans chaque ligne de la matrice correspondant à une contrainte au noeud i, les seuls éléments non nuls sont :

- les dérivées par rapport à U_i, θ_i, Q_i
- les dérivées par rapport à U_j et θ_j $\forall j \in e(i)$
- les dérivées par rapport à P_i, ou à P_{ij} $\forall j \in J_i$.

Exemple :

Considérons le problème 5 (voir annexe A1). Nous avons représenté par des croix les éléments non nuls de la matrice. (figure V.1.)

Soit n le nombre total de variables, N le nombre de noeuds, l le nombre de branches, p le nombre d'usines productrices, et $|e(o)|$ le nombre de noeuds reliés au noeud de référence.

On voit facilement que le nombre d'éléments non nuls de la matrice, non comprise la colonne correspondant à la variable μ (en effet cette colonne, n'étant composée que de 1, peut n'être pas enregistrée en mémoire) est égal à

$$n + 2N + 8l + p - 2|e(o)| - 1.$$

Pour le problème n°5, nous avons n = 50, N = 10, l = 13, p = 24, et $|e(o)| = 1$; le nombre d'éléments non nuls est donc de 195, ce qui correspond à une densité de 18,6 %.

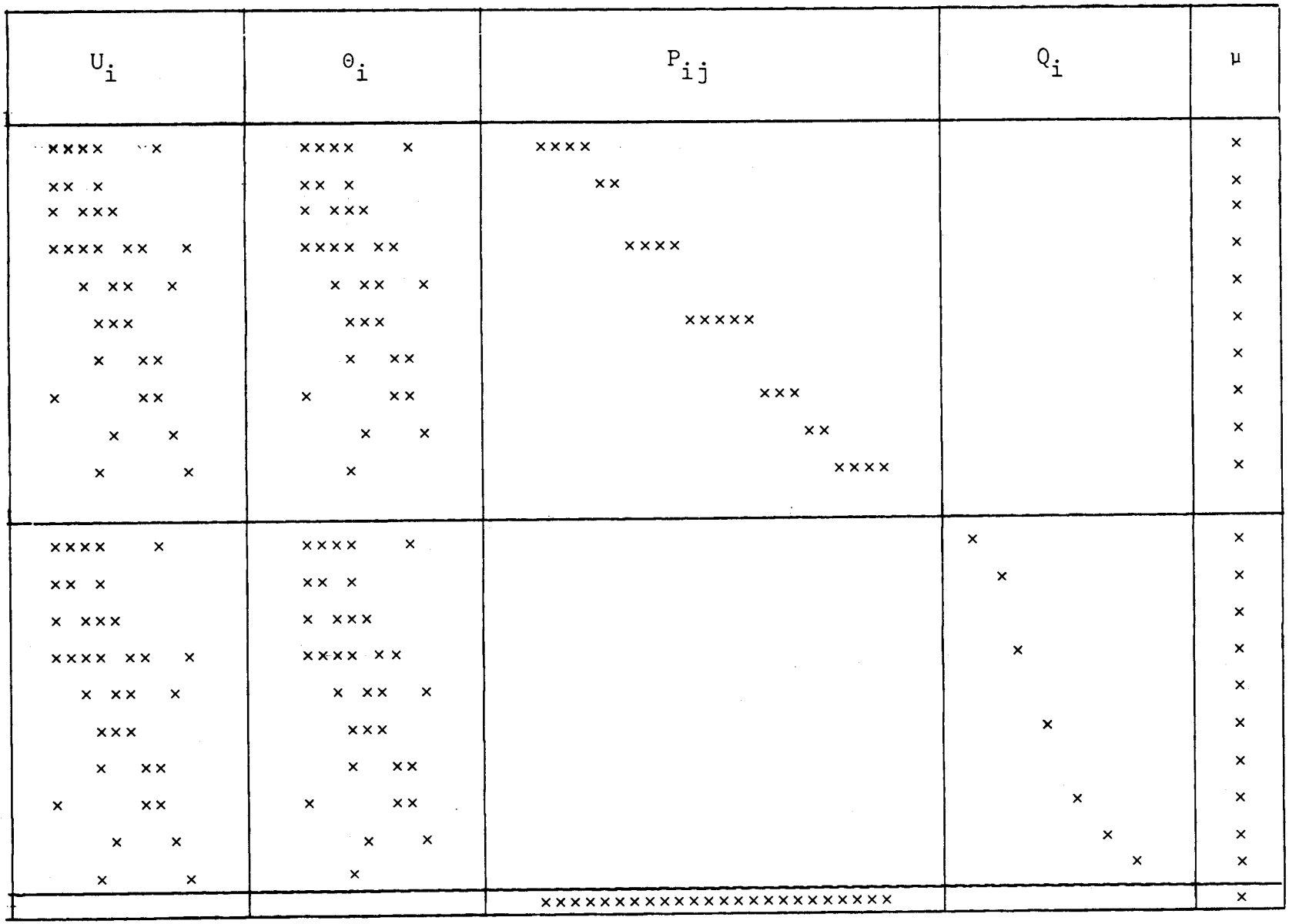


Figure V.1.

Pour le problème n°6, nous avons n = 103, N = 44, l = 49, p = 11, et |e(o)| = 4 : le nombre d'éléments non nuls est donc de 585, ce qui correspond à une densité de 6,4 %. Il est clair que la densité décroît au fur et à mesure que la taille du réseau augmente. Il est donc rentable de conserver A sous forme condensée ; nous utilisons pour cela le procédé décrit en II.4.3.

Remarquons que dans la partie de A relative aux U_i et θ_i nous avons 4 sous-matrices symétriques en structure (les matrices relatives aux θ_i ayant une colonne en moins puisque θ_{10} est fixé à 0). Mais ces sous-matrices ne sont pas symétriques en valeur, comme le montre le calcul suivant relatif à la matrice supérieure gauche :

Considérons la ligne i. Dans la dérivée par rapport à U_j le seul terme non nul est la dérivée de

$$-\frac{U_i U_j}{B_{ij}} \cos(\theta_i - \theta_j + \beta_{ij})$$

Soit :

$$-\frac{U_i}{B_{ij}} \cos(\theta_i - \theta_j + \beta_{ij}).$$

En considérant de même la ligne j, nous voyons que la dérivée par rapport à U_i est

$$-\frac{U_j}{B_{ij}} \cos(\theta_j - \theta_i + \beta_{ij})$$

Ces termes ne sont pas égaux.

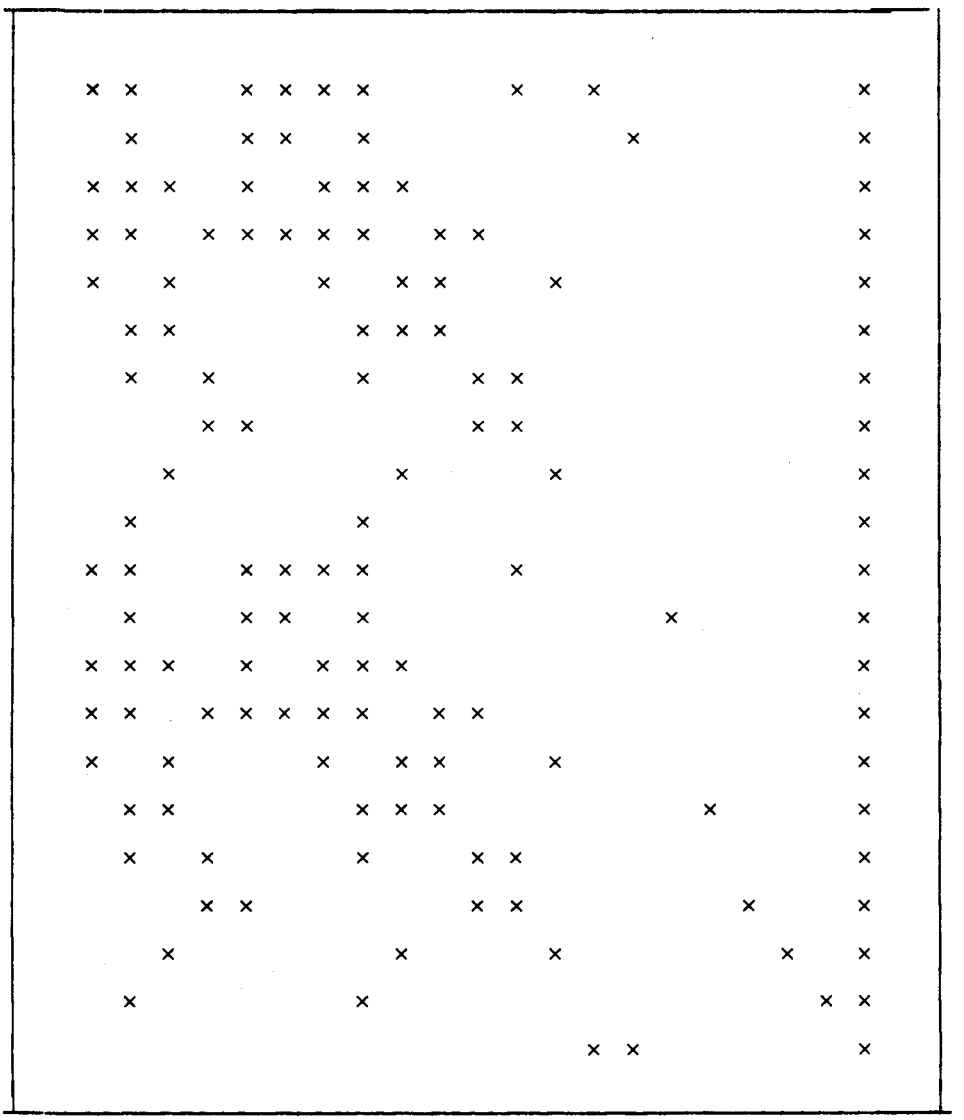
CALCULS DIRECTS.

Nous avons vu en II.2.7 que l'on doit résoudre pour une base donnée deux systèmes avec la matrice A^I et un système avec sa transposée.

A^I est constituée de colonnes prises dans la matrice A. Elle est donc, pour les gros problèmes, très creuse ; de plus, elle possède une structure particulière.

Exemple : considérons une matrice de base obtenue au cours de la résolution du problème n°5 de l'annexe A1.

(1)

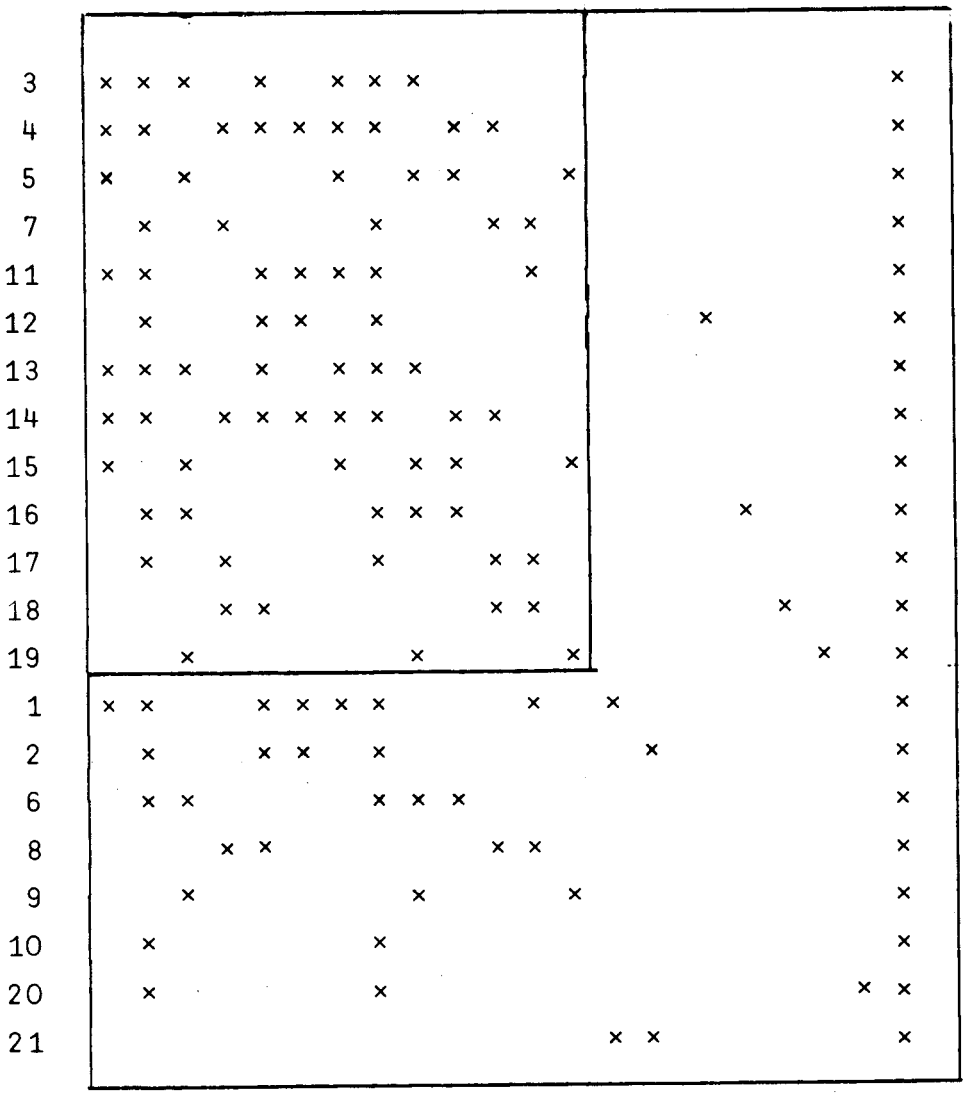


Par des échanges de lignes nous amenons sur la diagonale les termes correspondant à

$$\frac{\partial I_i(\theta,U)}{\partial U_i} \quad \text{et} \quad \frac{\partial K_i(\theta,U)}{\partial \theta_i} .$$

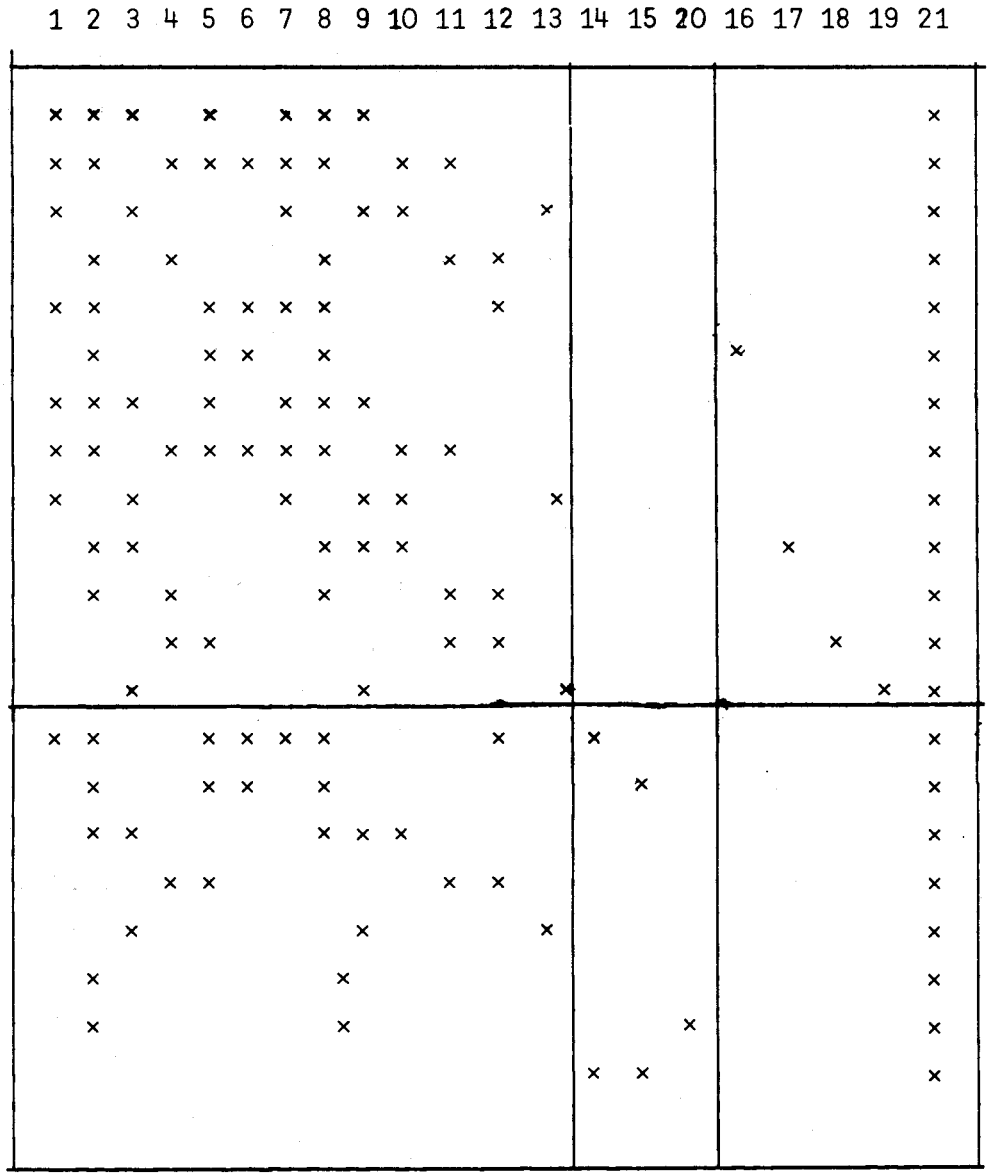
Nous obtenons ainsi un bloc symétrique en structure :

(2)

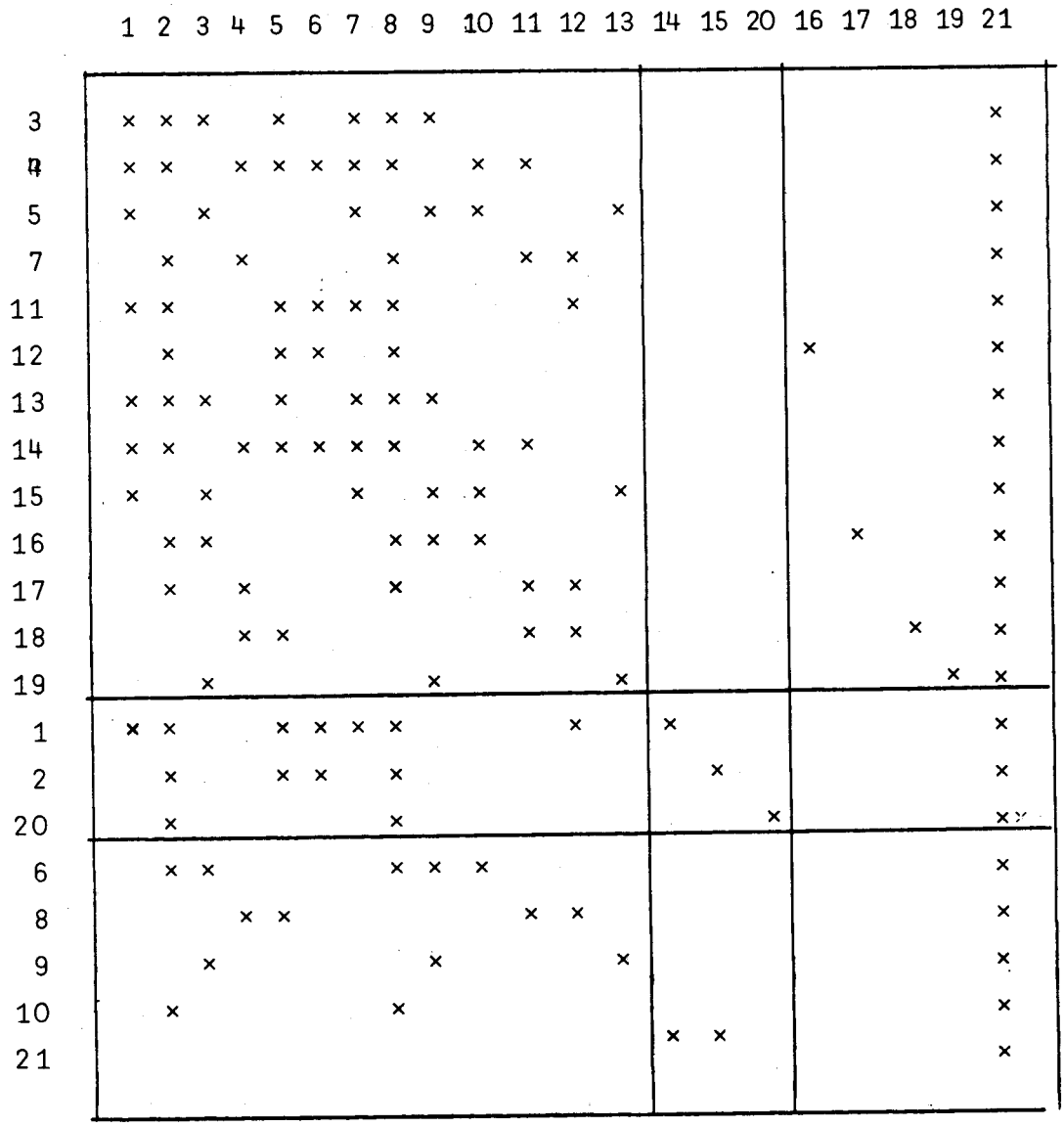


Puis par des échanges de colonnes nous créons à droite de ce bloc un bloc composé entièrement de zéros.

(3)



Enfin, par des échanges de lignes nous créons sous le bloc de zéros une matrice diagonale :



Quelle que soit la matrice de base A^I, ces opérations peuvent être effectuées. En pratique, les échanges de lignes et de colonnes sont effectués implicitement par utilisation d'adressage indirect. Finalement on obtient la décomposition

$$A^I = \begin{array}{|c|c|}\hline B & C \\ \hline D & E \\ \hline \end{array} \qquad \text{avec } B = \begin{array}{|c|c|}\hline M & O \\ \hline V & U \\ \hline \end{array}$$

M étant symétrique en structure et U diagonale.

Dans l'exemple précédent, A^I est une matrice d'ordre 21, B une matrice d'ordre 16 et M une matrice d'ordre 13.

Soit à résoudre le système

$$\begin{array}{|c|c|}\hline B & C \\ \hline D & E \\ \hline \end{array} \quad \begin{array}{|c|}\hline X \\ \hline Y \\ \hline \end{array} = \begin{array}{|c|}\hline U_o \\ \hline V_o \\ \hline \end{array}$$

A^I d'ordre n_3

B d'ordre n_2

$$\begin{cases} BX + CY = U_o \\ DX + EY = V_o \end{cases}$$

D'où en éliminant X

$$(DB^{-1}C - E)Y = DB^{-1}U_o - V_o$$

Posons $B^{-1}C = R$ et $B^{-1}U_o = T$

$\Longrightarrow$ $BR = C$ et $BT = U_o$.

Pour obtenir les $n_3 - n_2$ colonnes de R nous devons donc résoudre $n_3 - n_2$ systèmes et pour obtenir T un système, avec la matrice B.

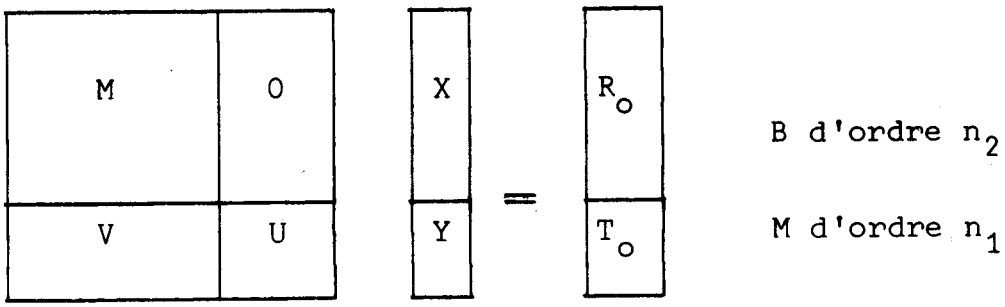
$$(DR - E)Y = DT - V_o$$

On obtient donc Y en résolvant un seul système avec une matrice d'ordre $n_3 - n_2$.

On en déduit $X = T - RY$.

Nous voyons donc que pour résoudre le système précédent on doit résoudre $n_3 - n_2 + 1$ systèmes avec la matrice B.

Considérons donc le système



$$\begin{cases} MX = R_o \\ VX + UY = T_o \end{cases}$$

X est obtenue par résolution du système $MX = R_o$.

On déduit Y par résolution de $UY = T_o - VX$, système dont la résolution est triviale puisque U est diagonale.
Finalement, on se ramène à $n_3 - n_2 + 1$ résolutions de systèmes avec la matrice M.

Remarque :

la résolution d'un système avec la transposée de A^I se décompose de la même manière et on se ramène à $n_3 - n_2 + 1$ résolutions de systèmes avec la transposée de M.

La matrice M étant creuse, nous la factorisons en utilisant l'élimination de Gauss, comme indiqué en II.2.4 :

$$M = LU$$

L, matrice triangulaire inférieure, étant factorisée en colonnes

$$L = L^{(1)}L^{(2)}\ldots L^{(n_1)}$$

U, matrice triangulaire supérieure, étant factorisée en lignes

$$U = U^{(n_1)}\,U^{(n_1 - 1)}\ldots U^{(1)}$$

En effet, M étant symétrique en structure, les matrices $L^{(k)}$ et $U^{(k)}$ ont une structure symétrique, ce qui permet un gain de place dans les procédés d'indiçage ([38]).

Pour minimiser le nombre d'éléments non nuls en mémoire, à chaque étape de l'élimination on choisit comme ligne du pivot la ligne la plus creuse de la matrice en cours de transformation (II.2.6)

La factorisation de M étant effectuée, la résolution des $n_3 - n_2 + 1$ systèmes se réduit à un minimum de calculs.

Remarque 1 :

La factorisation de M^t étant la transposée de la factorisation de M ; il suffit, pour résoudre les $n_3 - n_2 + 1$ systèmes avec M^t, d'échanger les rôles des $L^{(k)}$ et $U^{(k)}$; il n'est donc pas nécessaire de faire une seconde factorisation.

Remarque 2 :

Si, lors d'un changement de base de la méthode simpliciale, la matrice M n'est pas modifiée, c'est-à-dire si le changement de base ne porte pas sur les variables U_i ou θ_i, on n'a pas à refaire la factorisation, ce qui permet un gain en temps de calcul.

V.4.2 *Choix d'un point de départ* :

Nous avons vu en I.6.2 que, si l'on ne dispose pas d'un point réalisable, on peut en obtenir un facilement au bout d'un nombre fini de linéarisations à partir d'un point quelconque.

Toutefois on peut remarquer que, si les variables P_i et Q_i ne sont pas éliminées, la valeur des contraintes est d'autant plus grande, pour des U_i et θ_i donnés, que les P_i et Q_i ont une valeur plus élevée. On a donc intérêt à donner au départ à ces variables la valeur la plus grande possible, c'est-à-dire à les mettre à leur borne supérieure, pour obtenir un point plus proche du domaine.

V.4.3 *Changement de variables*

Dans la formulation du problème du Dispatching Economique, les variables θ_i n'interviennent que par les différences de la forme $\theta_i - \theta_j$, d'où le changement de variables classique :

$$\tau_{ij} = \theta_i - \theta_j \quad \forall i,j \text{ tels que } \exists \text{ une ligne entre } i \text{ et } j.$$

(on a donc $\tau_{ji} = - \tau_{ij}$).

Les nouvelles variables τ_{ij} sont relatives aux lignes du réseau, alors que les variables θ_i sont liées aux noeuds. Ce changement de variables simplifie les relations d'injection ; d'autre part il transforme les contraintes de limitations de transits

$$\theta_i - \theta_j \leq T_{ij} \qquad \theta_j - \theta_j \leq T_{ji}$$

en contraintes de bornes sur les τ_{ij} :

$$- T_{ij} \leqslant \tau_{ij} \leqslant T_{ij}.$$

Mais les variables τ_{ij} ne sont pas indépendantes. En effet, supposons qu'il y ait dans le graphe du réseau un cycle

$$i_1 i_2 i_3 \cdots i_{n-1} i_n i_1.$$

Nous devons avoir la relation

$$\tau_{i_1 i_2} + \tau_{i_2 i_3} + \cdots + \tau_{i_{n-1} i_n} + \tau_{i_n i_1} = 0.$$

Il s'introduit donc une contrainte linéaire en équation par cycle.

Une fois l'optimum du problème connu, il faudra alors calculer, à partir des τ_{ij}, les valeurs des θ_i, par une résolution d'un système d'équations linéaires.

Ce changement de variables est donc intéressant dans les problèmes comportant des contraintes de limitations de transits, à condition que le graphe du réseau correspondant comporte peu de cycles.

V.4.4 *Approximation des fonctions standard sinus et cosinus* :

Un procédé classique pour la résolution du problème du Dispatching Economique consiste à utiliser des développements limités à l'ordre 1 ou 2 des fontions envisagées. L'appel des fonctions standard sinus et cosinus étant fréquent, nous avons appliqué ce procédé à ces fonctions pour diminuer le temps de calcul.

1) Calcul d'erreur :

Cherchons d'abord à déterminer la manière dont les erreurs sur les sinus et cosinus interviennent dans le calcul des contraintes du problème, et donc sur la valeur de l'optimum.

Considérons la contrainte

$$P_i + \sum_{j \in e(i)} U_i U_j B'_{ij} \cos(\theta_i - \theta_j + \beta_{ij}) - \sum_{j \in e(i)} U_i^2 A_{ij} B'_{ij} \cos(\beta_{ij} - \alpha_{ij}) - C_i \geq 0.$$

où $B'_{ij} = \frac{1}{B_{ij}}$.

Nous supposons que les cosinus sont claculés avec une erreur E. L'erreur $\mathcal{E}$ qui en résulte sur la valeur de la contrainte est obtenue par différentiation :

$$\mathcal{E} \leq U_i \sum_{j \in e(i)} B'_{ij} |U_j - U_i A_{ij}| \times E.$$

Application numérique :

Considérons le problème n°5. Le calcul effectif sur ordinateur de la valeur ci-dessus, pour la 1ère contrainte de ce problème, donne

$$\mathcal{E} \leq 184\ E.$$

On voit que l'erreur commise sur la contrainte peut atteindre plusieurs centaines de fois l'erreur sur les fonctions standard.

2) Développement limité :

Nous avons tout d'abord déterminé, pour le problème n°5, les valeurs des angles dont le code de la Méthode des Centres par Linéarisations calcule les sinus et cosinus. La figure V.2 représente un résultat statistique obtenu sur près de 1200 valeurs de l'argument. On remarque que ces valeurs sont approximativement comprises entre 1,20 et 1,55.

Ces valeurs étant proches de $\frac{\pi}{2}$, il est logique d'effectuer les développements limités des fonctions sinus et cosinus au voisinage de $\frac{\pi}{2}$, de la manière suivante :

Soit α l'argument considéré, et $\theta = \frac{\pi}{2} - \alpha$.
Donc :

$$\sin \alpha = \cos \theta \qquad \cos \alpha = \sin \theta$$

avec θ proche de 0.
D'où

$$\sin \alpha = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} + \ldots + (-1)^n \frac{\theta^{2n}}{(2n)!}$$

$$\cos \alpha = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \ldots + (-1)^n \frac{\theta^{2n+1}}{(2n+1)!}$$

Le tableau de la figure V.3 donne, pour des arguments compris entre 1,13 et 1,56, la vraie valeur du sinus, puis successivement les valeurs du développement limité à 2,3 et 4 termes, avec l'erreur commise.

Le tableau de la figure V.4. donne les mêmes renseignements pour le cosinus.

Le tableau de la figure V.5. compare, pour les mêmes arguments, les vraies valeurs des sinus et cosinus avec les valeurs obtenues par un développement limité à 3 termes, avec correction sur le 3ème terme :

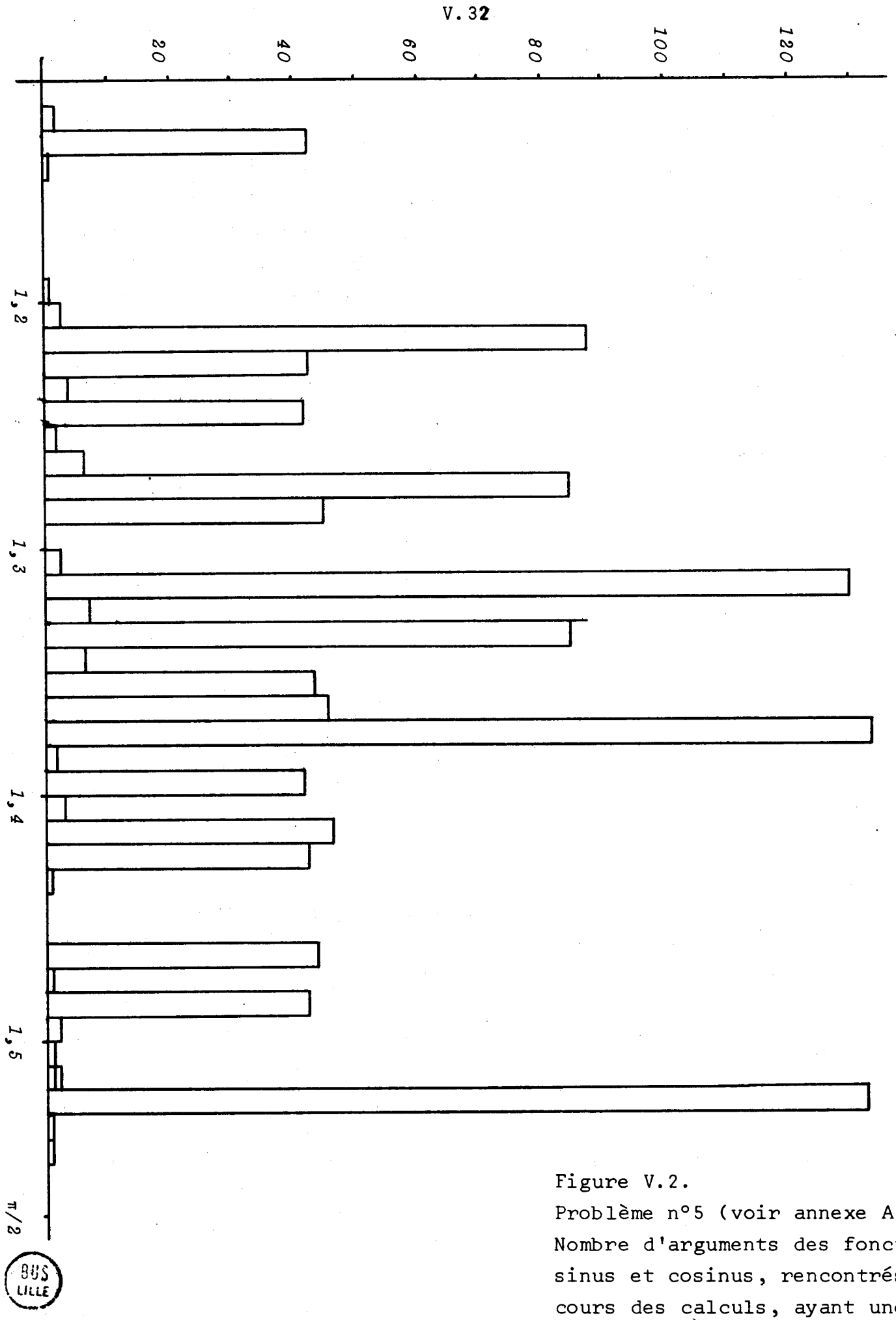


Figure V.2.
Problème n°5 (voir annexe A1)
Nombre d'arguments des fonctions sinus et cosinus, rencontrés au cours des calculs, ayant une valeur donnée.

θ	sin θ	2 termes		3 termes			
1.13	.904412189385	.902849299152	.16@-02	.904422342263	.10@-04	.904412154111	.35@-07
1.14	.908633496135	.907207262421	.14@-02	.908642344448	.88@-05	.908633466769	.29@-07
1.15	.912763940260	.911465225685	.13@-02	.912771626727	.77@-05	.912763915929	.24@-07
1.16	.916803108778	.915623188953	.12@-02	.916809763325	.67@-05	.916803088704	.20@-07
1.17	.920750597742	.919681152215	.11@-02	.920756338436	.57@-05	.920750581254	.16@-07
1.18	.924606012401	.923639115480	.97@-03	.924610946257	.49@-05	.924605998933	.13@-07
1.19	.928368967247	.927497078747	.87@-03	.928373191013	.42@-05	.928368956297	.11@-07
1.20	.932039085960	.931255042017	.78@-03	.932042686890	.36@-05	.932039077112	.88@-08
1.21	.935616001555	.934913005282	.70@-03	.935619058097	.31@-05	.935615994439	.71@-08
1.22	.939099356324	.938470968550	.63@-03	.939101938837	.26@-05	.939099350641	.57@-08
1.23	.942488801923	.941928931813	.56@-03	.942490973309	.22@-05	.942488797419	.45@-08
1.24	.945783999438	.945286895078	.50@-03	.945785815721	.18@-05	.945783995887	.36@-08
1.25	.948984619349	.948544858346	.44@-03	.948986130281	.15@-05	.948984616569	.28@-08
1.26	.952090341586	.951702821617	.39@-03	.952091591188	.12@-05	.952090339425	.22@-08
1.27	.955100855574	.954760784882	.34@-03	.955101882648	.10@-05	.955100853915	.17@-08
1.28	.958015860279	.957718748151	.30@-03	.958016698862	.84@-06	.958015859013	.13@-08
1.29	.960835064189	.960576711414	.26@-03	.960835744025	.68@-06	.960835063229	.96@-09
1.30	.963558185404	.963334674680	.22@-03	.963558732360	.55@-06	.963558184684	.72@-09
1.31	.966184951600	.965992637949	.19@-03	.966185388063	.44@-06	.966184951069	.53@-09
1.32	.968715100108	.968550601220	.16@-03	.968715445337	.35@-06	.968715099722	.39@-09
1.33	.971148377913	.971008564486	.14@-03	.971148648374	.27@-06	.971148377629	.28@-09
1.34	.973484541670	.973366527748	.12@-03	.973484751385	.21@-06	.973484541474	.20@-09
1.35	.975723357808	.975624491019	.99@-04	.975723518592	.16@-06	.975723357670	.14@-09
1.36	.977864602416	.977782454286	.82@-04	.977864724179	.12@-06	.977864602321	.95@-10
1.37	.979908061376	.979840417555	.68@-04	.979908152348	.91@-07	.979908061311	.65@-10
1.38	.981853530352	.981798380819	.55@-04	.981853597312	.67@-07	.981853530308	.44@-10
1.39	.983700814798	.983656344094	.44@-04	.983700863277	.48@-07	.983700814768	.29@-10
1.40	.985449729963	.985414307356	.35@-04	.985449764429	.34@-07	.985449729948	.15@-10
1.41	.987100101006	.987072270628	.28@-04	.987100124995	.24@-07	.987100100991	.15@-10
1.42	.988651762828	.988630233895	.22@-04	.988651779155	.16@-07	.988651762821	.73@-11
1.43	.990104560328	.990088197165	.16@-04	.990104571140	.11@-07	.990104560321	.73@-11
1.44	.991458348168	.991446160430	.12@-04	.991458355124	.70@-08	.991458348168	.00@+00
1.45	.992712991024	.992704123705	.89@-05	.992712995338	.43@-08	.992712991024	.00@+00
1.46	.993868363398	.993862086968	.63@-05	.993868365966	.26@-08	.993868363398	.00@+00
1.47	.994924349764	.994920050240	.43@-05	.994924351219	.15@-08	.994924349764	.00@+00
1.48	.995880844525	.995878013508	.28@-05	.995880845304	.78@-09	.995880844525	.00@+00
1.49	.996737752032	.996735976779	.18@-05	.996737752418	.39@-09	.996737752032	.00@+00
1.50	.997494986589	.997493940044	.10@-05	.997494986763	.17@-09	.997494986589	.00@+00
1.51	.998152472481	.998151903312	.57@-06	.998152472554	.73@-10	.998152472481	.00@+00
1.52	.998710143972	.998709866583	.28@-06	.998710143993	.22@-10	.998710143972	.00@+00
1.53	.999167945267	.999167829856	.12@-06	.999167945275	.73@-11	.999167945267	.00@+00
1.54	.999525830603	.999525793125	.37@-07	.999525830603	.00@+00	.999525830603	.00@+00
1.55	.999783764188	.999783756396	.78@-08	.999783764188	.00@+00	.999783764188	.00@+00
1.56	.999941720229	.999941719662	.57@-09	.999941720229	.00@+00	.999941720229	.00@+00

FIGURE-V.3

θ	cos θ	2 termes		3 termes		4 termes	
1.13	.426659807912	.426521769410	.14@-03	.426660447736	.64@-06	.426659806183	.17@-08
1.14	.417594503937	.417471403273	.12@-03	.417595048889	.54@-06	.417594502534	.14@-08
1.15	.408487440866	.408377957507	.11@-03	.408487903256	.46@-06	.408487439731	.11@-08
1.16	.399339529400	.399242432106	.97@-04	.399339920186	.39@-06	.399339528483	.92@-09
1.17	.390151684306	.390065827072	.86@-04	.390152013208	.33@-06	.390151683570	.73@-09
1.18	.380924824372	.380849142404	.76@-04	.380925099985	.28@-06	.380924823787	.59@-09
1.19	.371659872270	.371593378105	.66@-04	.371660102173	.23@-06	.371659871804	.47@-09
1.20	.362357754487	.362299534172	.58@-04	.362357945335	.19@-06	.362357754123	.36@-09
1.21	.353019401236	.352968610607	.51@-04	.353019558859	.16@-06	.353019400953	.28@-09
1.22	.343645746339	.343601607411	.44@-04	.343645875822	.13@-06	.343645746115	.22@-09
1.23	.334237727153	.334199524580	.38@-04	.334237832916	.11@-06	.334237726981	.17@-09
1.24	.324796284470	.324763362115	.33@-04	.324796370338	.86@-07	.324796284339	.13@-09
1.25	.315322362433	.315294120024	.28@-04	.315322431706	.69@-07	.315322362337	.98@-10
1.26	.305816908419	.305792798293	.24@-04	.305816963926	.56@-07	.305816908346	.73@-10
1.27	.296280872973	.296260396935	.20@-04	.296280917123	.44@-07	.296280872920	.55@-10
1.28	.286715209687	.286697915938	.17@-04	.286715244533	.35@-07	.286715209647	.40@-10
1.29	.277120875113	.277106355314	.15@-04	.277120902395	.27@-07	.277120875084	.29@-10
1.30	.267498828688	.267486715054	.12@-04	.267498849853	.21@-07	.267498828665	.22@-10
1.31	.257850032599	.257839995162	.10@-04	.257850048865	.16@-07	.257850032584	.15@-10
1.32	.248175451727	.248167195636	.83@-05	.248175464099	.12@-07	.248175451715	.11@-10
1.33	.238476053514	.238469316481	.67@-05	.238476062819	.93@-08	.238476053506	.73@-11
1.34	.228752807891	.228747357692	.55@-05	.228752814809	.69@-08	.228752807888	.55@-11
1.35	.219006687184	.219002319269	.44@-05	.219006692257	.51@-08	.219006687181	.36@-11
1.36	.209238665987	.209235201214	.35@-05	.209238669654	.37@-08	.209238665984	.18@-11
1.37	.199449721099	.199447003529	.27@-05	.199449723710	.26@-08	.199449721098	.18@-11
1.38	.189640831405	.189638726209	.21@-05	.189640833229	.18@-08	.189640831403	.18@-11
1.39	.179812977783	.179811369256	.16@-05	.179812979036	.13@-08	.179812977783	.00@+00
1.40	.169967143017	.169965932671	.12@-05	.169967143857	.84@-09	.169967143017	.00@+00
1.41	.160104311678	.160103416453	.90@-06	.160104312229	.55@-09	.160104311678	.00@+00
1.42	.150225470040	.150224820603	.65@-06	.150225470391	.35@-09	.150225470040	.00@+00
1.43	.140331605980	.140331145118	.46@-06	.140331606198	.22@-09	.140331605980	.00@+00
1.44	.130423708877	.130423390002	.32@-06	.130423709005	.13@-09	.130423708877	.00@+00
1.45	.120502769511	.120502555255	.21@-06	.120502769587	.75@-10	.120502769511	.00@+00
1.46	.110569779970	.110569640873	.14@-06	.110569780011	.41@-10	.110569779970	.00@+00
1.47	.100625733543	.100625646859	.87@-07	.100625733564	.21@-10	.100625733543	.00@+00
1.48	.090671624625	.090671573212	.51@-07	.090671624635	.10@-10	.090671624625	.00@+00
1.49	.080708448620	.080708419933	.29@-07	.080708448624	.45@-11	.080708448620	.00@+00
1.50	.070737201840	.070737187020	.15@-07	.070737201841	.18@-11	.070737201840	.00@+00
1.51	.060758881396	.060758874475	.69@-08	.060758881397	.45@-12	.060758881396	.00@+00
1.52	.050774485115	.050774482297	.28@-08	.050774485115	.00@+00	.050774485115	.00@+00
1.53	.040785011429	.040785010487	.94@-09	.040785011429	.00@+00	.040785011429	.00@+00
1.54	.030791459275	.030791459044	.23@-09	.030791459275	.00@+00	.030791459275	.00@+00
1.55	.020794828001	.020794827969	.33@-10	.020794828001	.00@+00	.020794828001	.00@+00
1.56	.010796117262	.010796117260	.13@-11	.010796117262	.00@+00	.010796117262	.00@+00

FIGURE V.4

θ	sin θ	3 termes avec correction		cos θ	3 termes avec correction	
1.13	.904412189385	.904415279302	.31@-05	.426659807912	.426659887481	.80@-07
1.14	.908633496135	.908635900926	.24@-05	.417594503937	.417594549362	.45@-07
1.15	.912763940260	.912765760988	.18@-05	.408487440866	.408487459080	.18@-07
1.16	.916803108778	.916804435607	.13@-05	.399339529400	.399339526332	.31@-08
1.17	.920750597742	.920751510845	.91@-06	.390151684306	.390151665016	.19@-07
1.18	.924606012401	.924606582742	.57@-06	.380924824372	.380924793114	.31@-07
1.19	.928368967247	.928369257267	.29@-06	.371659872270	.371659832604	.40@-07
1.20	.932039085960	.932039150367	.64@-07	.362357754487	.362357709356	.45@-07
1.21	.935616001555	.935615887919	.11@-06	.353019401236	.353019353031	.48@-07
1.22	.939099356324	.939099105776	.25@-06	.343645746339	.343645696979	.49@-07
1.23	.942488801923	.942488449737	.35@-06	.334237727153	.334237678151	.49@-07
1.24	.945783999438	.945783575563	.42@-06	.324796284470	.324796236984	.47@-07
1.25	.948984619349	.948984148965	.47@-06	.315322362433	.315322317328	.45@-07
1.26	.952090341586	.952089845613	.50@-06	.305816908419	.305816866297	.42@-07
1.27	.955100855574	.955100351118	.50@-06	.296280872973	.296280834221	.39@-07
1.28	.958015860279	.958015361061	.50@-06	.286715209687	.286715174524	.35@-07
1.29	.960835064189	.960834580971	.48@-06	.277120875113	.277120843625	.31@-07
1.30	.963558185404	.963557726342	.46@-06	.267498828688	.267498800827	.28@-07
1.31	.966184951600	.966184522610	.43@-06	.257850032599	.257850008248	.24@-07
1.32	.968715100108	.968714705183	.39@-06	.248175451727	.248175430693	.21@-07
1.33	.971148377913	.971148019397	.36@-06	.238476053514	.238476035563	.18@-07
1.34	.973484541670	.973484220568	.32@-06	.228752807891	.228752792763	.15@-07
1.35	.975723357808	.975723073958	.28@-06	.219006687184	.219006674588	.13@-07
1.36	.977864602416	.977864354786	.25@-06	.209238665987	.209238655641	.10@-07
1.37	.979908061376	.979907848220	.21@-06	.199449721099	.199449712719	.84@-08
1.38	.981853530352	.981853349384	.18@-06	.189640831405	.189640824716	.67@-08
1.39	.983700814798	.983700663385	.15@-06	.179812977783	.179812972532	.52@-08
1.40	.985449729963	.985449605224	.12@-06	.169967143017	.169967138965	.41@-08
1.41	.987100101006	.987099999928	.10@-06	.160104311678	.160104308609	.31@-08
1.42	.988651762828	.988651682421	.80@-07	.150225470040	.150225467765	.23@-08
1.43	.990104560328	.990104497616	.63@-07	.140331605980	.140331604336	.16@-08
1.44	.991458348168	.991458300373	.48@-07	.130423708877	.130423707717	.12@-08
1.45	.992712991024	.992712955510	.36@-07	.120502769511	.120502768721	.79@-09
1.46	.993868363398	.993868337772	.26@-07	.110569779970	.110569779449	.52@-09
1.47	.994924349764	.994924331908	.18@-07	.100625733543	.100625733213	.33@-09
1.48	.995880844525	.995880832585	.12@-07	.090671624625	.090671624427	.20@-09
1.49	.996737752032	.996737744451	.76@-08	.080708448620	.080708448509	.11@-09
1.50	.997494986589	.997494982070	.45@-08	.070737201840	.070737201782	.58@-10
1.51	.998152472481	.998152470000	.25@-08	.060758881396	.060758881368	.28@-10
1.52	.998710143972	.998710142742	.12@-08	.050774485115	.050774485103	.11@-10
1.53	.999167945267	.999167944758	.51@-09	.040785011429	.040785011425	.36@-11
1.54	.999525830603	.999525830436	.17@-09	.030791459275	.030791459274	.91@-12
1.55	.999783764188	.999783764152	.36@-10	.020794828001	.020794828001	.23@-12
1.56	.999941720229	.999941720222	.73@-11	.010796117262	.010796117262	.00@+00

FIGURE V.5

$$\sin \alpha = 1 - \frac{\theta^2}{2!} + 0{,}99551 \frac{\theta^4}{4!}$$

$$\cos \alpha = \theta - \frac{\theta^3}{3!} + 0{,}99596 \frac{\theta^5}{5!}$$

Les valeurs des coefficients correcteurs ayant été déterminées empiriquement de manière à minimiser l'erreur maximum sur l'intervalle considéré.

Ces résultats, ainsi que le calcul d'erreur précédent, montrent que si l'on désire sur les contraintes une précision de l'ordre de 10^{-5}, il est nécessaire de prendre au minimum 4 termes du développement limité.

3) Temps de calcul

Nous avons comparé les temps mis par l'ordinateur pour effectuer les calculs précédents.

Temps d'appel de la fonction standard sinus : $5{,}8.10^{-3}$s.

Temps du développement limité à 3 termes, avec correction sur le 3ème terme : $1{,}8.10^{-3}$s.

Temps du développement limité à 4 termes : $1{,}9.10^{-3}$s.

Temps d'appel de la fonction standard cosinus : $5{,}6.10^{-3}$s.

Temps du développement limité à 3 termes, avec correction sur le 3ème terme : $2{,}0.10^{-3}$s.

Temps du développement limité à 4 termes : $2{,}4.10^{-3}$s.

Ces temps sont des moyennes calculées sur l'appel de 4300 fonctions.

On voit donc que, si l'on ne désire pas une forte précision sur le calcul des contraintes, et donc sur la solution du problème, on gagnera du temps en se contentant de 4 termes du développement limité des fonctions standard.

CHAPITRE VI

EXPERIENCES NUMERIQUES COMPARATIVES

VI.1 Introduction :

Des expériences numériques systématiques ont été effectuées sur les problèmes cités en annexe A1, à l'aide d'un code écrit en ALGOL 60.

Pour les problèmes-tests 1 à 5, ces expériences numériques ont été réalisées sur Gamma M40 Bull. Les principales caractéristiques de cet ordinateur sont :

- Unité de mémoire de 32768 mots de 24 bits
- Durée du cycle de la mémoire fixe : 1,33 µs
- Durée du cycle de la mémoire principale : 4,4µs
- Opérations arithmétiques incorporées en mémoire fixe
  Virgule fixe simple longueur (24 bits) :

  addition 10µs

  multiplication 55µs

  Virgule flottante (48 bits, mantisse sur 37 bits) :

  addition 80µs

  multiplication 185µs

Les expériences numériques concernant le problème n°6, de grande taille, ont été réalisées sur 10070 CII.
Les caractéristiques générales de cet ordinateur sont :

- Capacité mémoire de 98304 mots de 32 bits par blocs
- Durée du cycle actif de 755 à 1 230 ns
- Durée de sélection de 25 à 330 ns

  } cycle mémoire

- Recouvrement possible dans le temps de cycles effectués dans des blocs mémoire différents.
- Opérations arithmétiques
  Addition double flottante (64bits)

sans opérandes normalisés : de 13,7 à 15,1 µs
avec opérandes normalisés : de 4,1 à 5,5 µs

Division double flottante (64 bits) :

sans opérandes normalisés : de 34,7 à 36,3 µs
avec opérandes normalisés ≠o: de 25,4 à 27,0 µs

Multiplication double flottante (64 bits) :

sans opérandes normalisés : de 16,6 à 18,0 µs
avec opérandes normalisés ≠o: de 9,1 à 10,6 µs

Il faut ajouter à ces valeurs un temps additionnel de 0,3 µs à 2,3 µs suivant le registre utilisé.

Les résultats obtenus montrent que les problèmes traités sont résolus par la méthode des centres linéarisée avec une excellente précision, les valeurs des contraintes actives à l'optimum trouvé étant très faibles.

L'algorithme de centrage (I.5) accélère sensiblement la convergence pratique de la méthode ; on constate que le choix n°2 de I.5.3 est sur certains problèmes plus efficace que le choix n°1.
La pondération de la fonction économique (I.6.1) donne également de bons résultats ; la meilleure valeur du coefficient de pondération à choisir semble dépendre davantage de la forme du modèle que de la taille du problème. La non-continuité du gradient de la fonction économique de certains problèmes produit de graves ennuis et on a toujours intérêt à présenter ces problèmes sous une forme dans laquelle la fonction économique est linéaire (V.3.2). Lorsque les conditions optimales de résolution sont réunies, une bonne valeur de l'optimum est obtenue généralement en un nombre de troncatures compris entre 10 et 15.

Dans ce qui suit, nous utiliserons les paramètres de programmation suivants :

k pondération de la fonction économique (I.6.1)

lin nombre de linéarisations par toncature (I.4.2)
dic réduction de l'intervalle dans la dichotomie (III.2)

VI.2 **Problème n°1** (voir annexe A1)

Dans ce problème, la fonction économique est continuement différentiable.

Après réduction du nombre de contraintes en équation (voirV.3.1), ce problème comporte :

- 2 variables θ_1 et θ_2
- 5 contraintes non linéaires, dont une en équation
- 2 contraintes linéaires
- 4 contraintes de bornes (2 sur chaque variable).

La figure VI.1 représente les contraintes du problème. On voit que l'optimum se trouve sur une seule contrainte, qui est bien entendu la contrainte non linéaire en équation.

Cette contrainte est transformée en inéquation par mise en $\leq$.
La figure VI.2 montre le domaine réalisable, ainsi que quelques équipotentielles de la fonction économique. Le point de départ est le point obtenu en mettant les variables à leur borne inférieure.

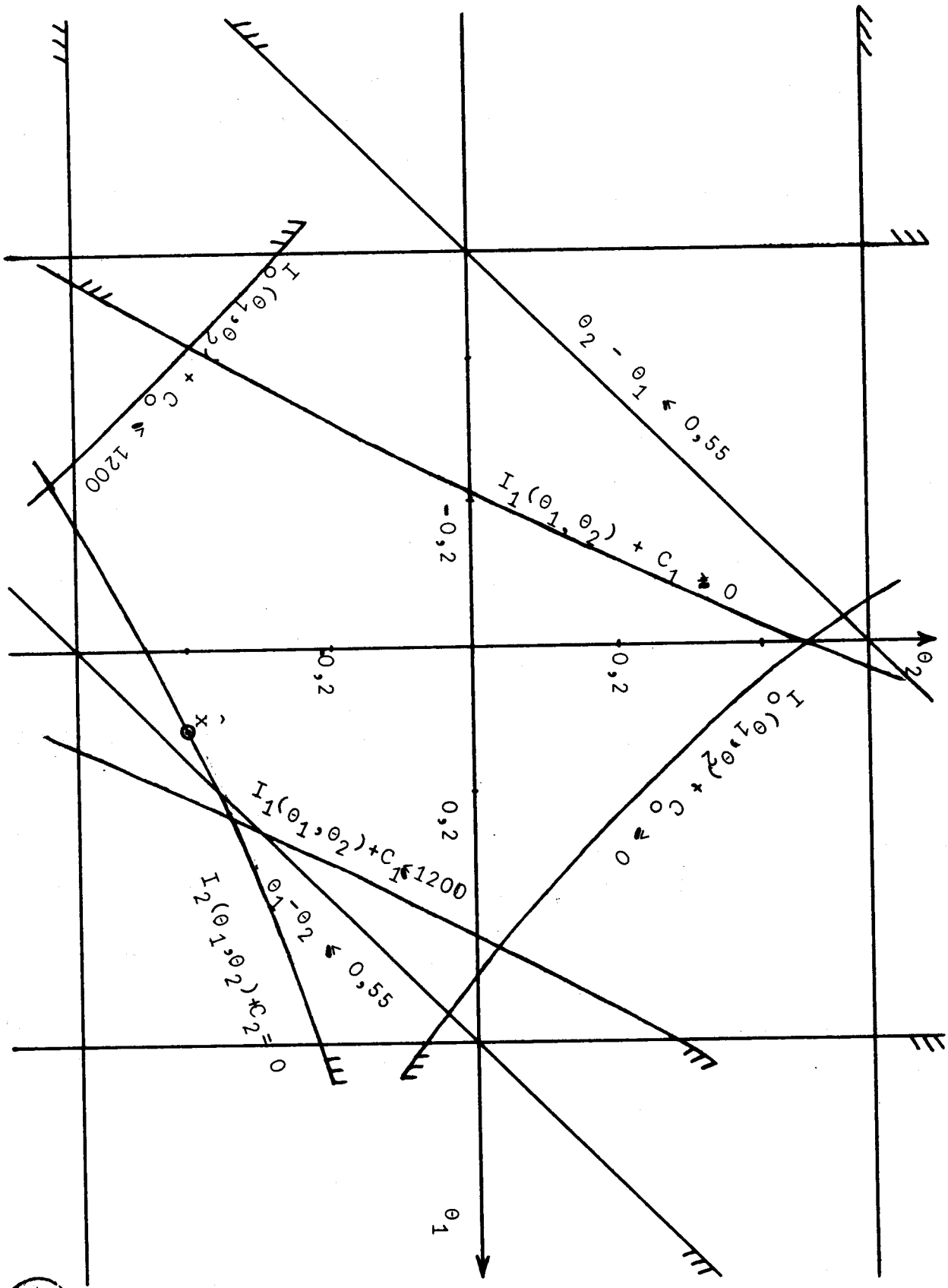


Figure VI.1
Problème N°1(voir annexe A1)
Les contraintes du problème

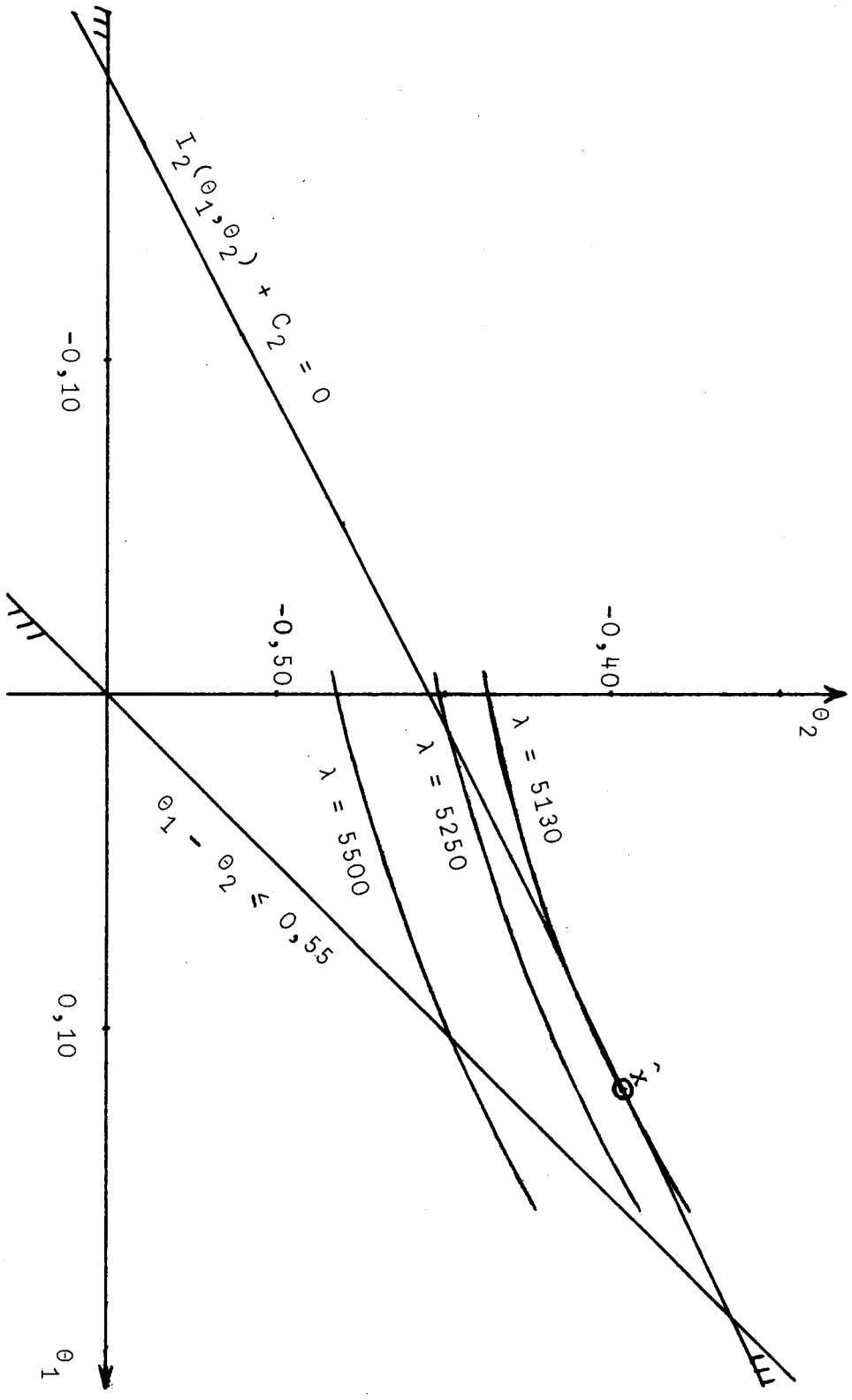


Figure VI.2
Problème N°1 (voir annexe A1)
le domaine réalisable

1) Essais sans algorithme de centrage :

Paramètres de programmation :

lin = 3 dic = 10^4.

Dans le tableau suivant nous donnons la valeur de la fonction économique en fonction du numéro de troncature, pour 2 valeurs différentes du coefficient de pondération de la fonction économique.

Troncature	k=1	$k=10^{-2}$
0	8880	8880
1	6111,4	5191,2
2	5625,9	5139,2
4	5261,0	5133,0
6	5195,9	5130,9
8	5175,0	5129,8
10	5164,1	5129,1
15	5150,9	5128,2
20	5144,6	5127,7
25	5140,9	5127,4
30	5138,5	5127,2
35		5127,1

La convergence est lente, mais on voit que la pondération $k=10^{-2}$ l'accélère.

Cette lenteur de convergence est principalement due à de fortes oscillations sur le point (voir annexe A3.1.1)

2) <u>essais avec algorithme de centrage</u> (voir I.5)

Les 2 contraintes linéaires sont mises dans la partie non linéaire ; elles interviennent donc dans la définition de la F-distance (voir I.4)

Paramètre de programmation :

lin = 3 dic = 10^4.

Centrage (choix n°1 de I.5.3)

Le tableau suivant donne la valeur de la fonction économique en fonction du numéro de troncature, pour 2 valeurs différentes de k.

Troncature	$k=10^{-1}$	$k=10^{-2}$
0	7596,75	7596,75
2	5149,32	5141,57
4	5126,39	5129,96
6	5126,20	5127,10
8		5126,41
10		5126,25
12		5126,21
14		5126,20

Le meilleur résultat est obtenu avec $k=10^{-1}$.

3) Meilleure solution obtenue

Le meilleur essai est l'essai avec centrage et avec pour paramètres de programmation k = 10^{-1}, lin = 3, dic = 10^{4}.

Dans ces conditions :
La valeur 5126,200 (arrondi aux 3 premières décimales) est obtenue à la 7ème troncature.

La meilleure valeur obtenue est

5126,20024854

à la 11ème troncature, ce qui correspond à la solution

θ_1 = 0,118893 P_0 = 679,920548

θ_2 = -0,396226 P_1 =1026,093603

La variation de la valeur de la fonction économique en fonction du numéro de troncature dans cet essai a été représentée par la figure VI.3 (voir les valeurs numériques en A3.1.2).

VI.3 Problème n°2 (voir annexe A1)

Seule la partie linéaire du problème précédent est modifiée. Donc la fonction économique est continuement différentiable, et après réduction du nombre de contraintes en équation, ce problème comporte :

- 2 variables θ_1 et θ_2
- 5 contraintes non linéaires, dont une en équation
- 2 contraintes linéaires
- 4 contraintes de bornes (2 sur chaque variable).

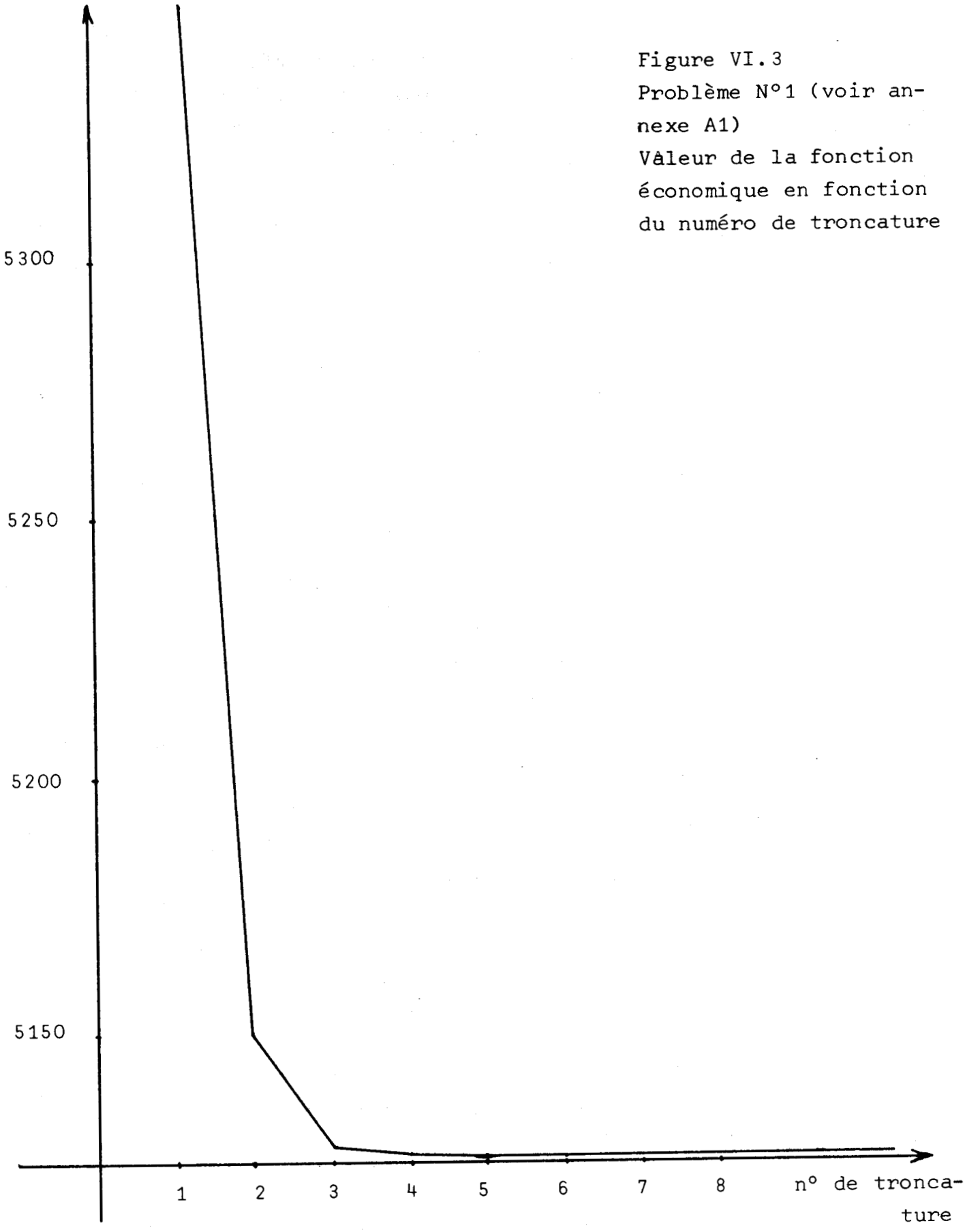


Figure VI.3
Problème N°1 (voir annexe A1)
Valeur de la fonction économique en fonction du numéro de troncature

La contrainte en équation est transformée en inéquation par mise en $\leq$. La figure VI.4 montre le domaine réalisable. On voit que l'optimum se trouve à l'intersection de deux contraintes : la contrainte non linéaire en équation, et une contrainte linéaire.

Le point de départ est le point obtenu en mettant les variables à leur borne inférieure.

1) Essais :

Paramètres de programmation :

lin = 3 dic = 10^4

Pas d'algorithme de centrage.

Valeur de la fonction économique en fonction du numéro de troncature :

Troncature	k=1	$k=10^{-2}$
0	8880	8880
2	5366,82	5174,86
4	5242,03	5174,13
6	5197,65	
8	5182,22	
10	5176,91	
14	5174,45	
18	5174,16	
22	5174,13	

La convergence est rapide lorsque $k=10^{-2}$.

L'algorithme de centrage est donc inutile pour ce problème.

2) Meilleure solution obtenue

Le meilleur essai est l'essai avec pour paramètres de programmation $k=10^{-2}$, lin = 3, dic = 10^{4}.

Dans ces conditions :

La valeur 5174,126 (arrondi aux 3 premières décimales) est obtenue à la 4ème troncature.

La meilleure valeur obtenue est

5174,126061

à la 6ème troncature, ce qui correspond à la solution

θ_1 = 0,051109 P_0 = 776,159027

θ_2 = -0,428891 P_1 = 925,195138

En ce point la contrainte en équation vaut 15.10^{-9}. (voir pour cet essai les valeurs numériques en A3.2).

VI.4 **Problème n°3** (voir annexe A1)

Dans ce problème, la fonction économique est continuement différentiable.

Après réduction du nombre de contraintes en équation (voir V.3.1), ce problème comporte :

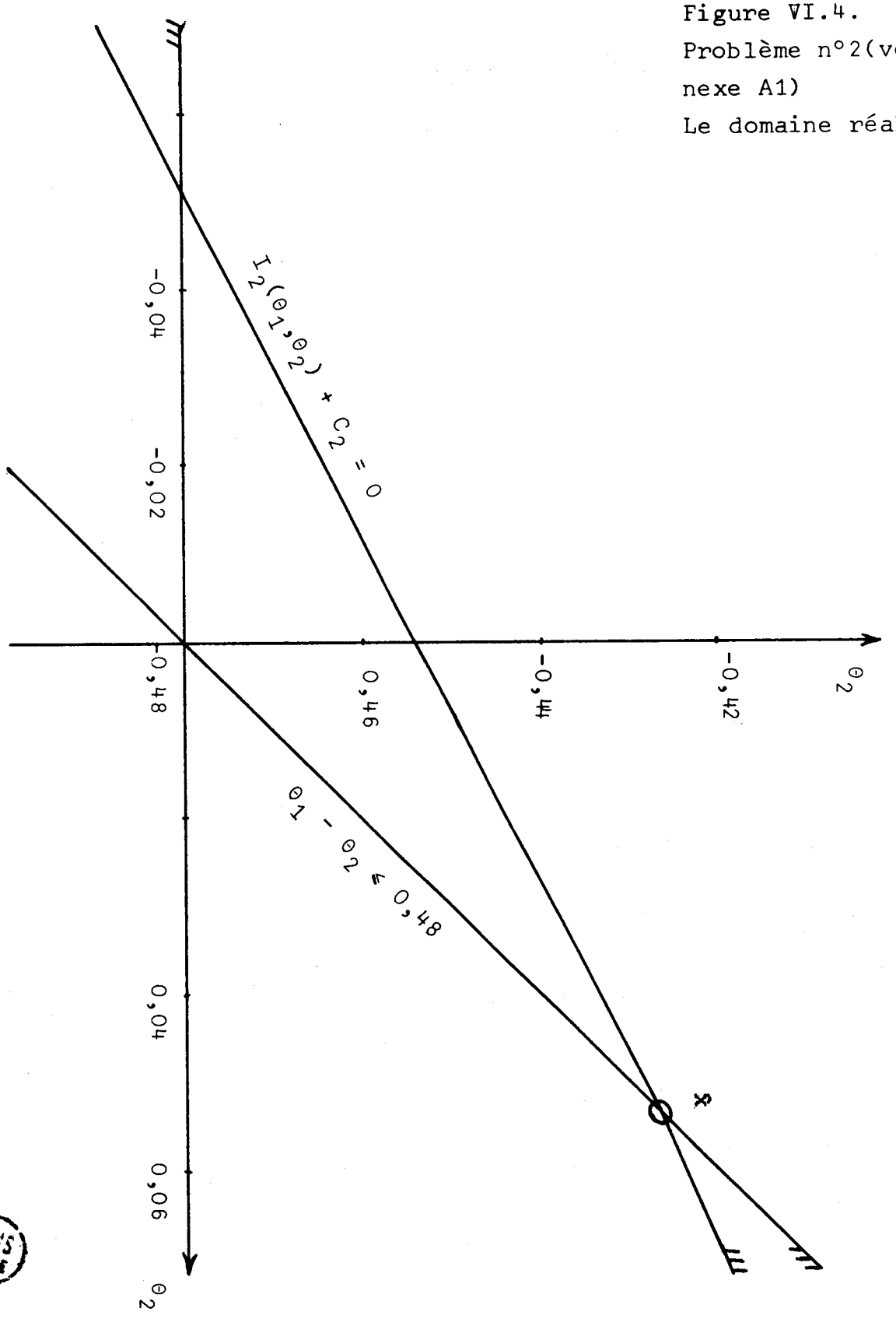


Figure VI.4.
Problème n°2(voir annexe A1)
Le domaine réalisable

- 5 variables U_o, U_1, U_2, θ_1 et θ_2

- 10 contraintes non linéaires, dont deux en équation

- 2 contraintes linéaires

- 10 contraintes de bornes (2 sur chaque variable).

Les deux contraintes non linéaires en équation sont transformées en inéquations par mise en $\leq$.

Le point de départ est le point obtenu en mettant les variables à leur borne inférieure.

1) Essais sans algorithme de centrage

Paramètres de programmation :

lin = 3 dic = 10^4

Valeur de la fonction économique en fonction du numéro de troncature

Troncature	k=1	$k=10^{-2}$
0	10000	10000
5	5455,64	5372,44
10	5423,36	5367,55
15	5408,97	5365,73
20		5364,78
25		5364,21
30		5363,82
40		5363,33
50		5363,02

La convergence est très lente, bien que la pondération $k=10^{-2}$ l'accélère (voir détails en A3.3.1).

Cette lenteur est principalement due à de fortes oscillations sur le point (voir A3.3.2).

2) Essais avec algorithme de centrage (voir I.5)

Les 2 contraintes linéaires sont mises dans la partie non linéaire ; elles interviennent donc dans la définition de la F-distance (voir I.4.)

Paramètres de programmation :

lin = 1 dic = 10^9 k = 10^{-2}

3 contraintes additionnelles. (choix n°1 de I.5.3)

Valeur de la fonction économique en fonction du numéro de troncature :

0	5640,953
1	5395,677
2	5367,128
3	5362,595
4	5361,781
5	5361,762
6	5361,761

Dans ce cas, l'optimum est atteint.

3) Meilleure solution obtenue

Le meilleur essai est l'essai avec centrage et avec pour paramètre de programmation $k=10^{-2}$, lin = 1, dic = 10^{9}.

Dans ces conditions :

La valeur 5361,761 (arrondi aux 3 premières décimales) est obtenue à la 6ème troncature.

La meilleure valeur obtenue est

5361,76140909

à la 8ème troncature, ce qui correspond à la solution

$U_0 = 252$ $\qquad$ $P_0 = 675{,}000377$

$U_1 = 252$ $\qquad$ $P_1 = 1134{,}048246$

$U_2 = 201{,}465617$ $\qquad$ $Q_0 = 426{,}626602$

$\Theta_1 = 0{,}133500$ $\qquad$ $Q_1 = 368{,}489268$

$\Theta_2 = -0{,}371183$

En ce point les 2 contraintes en équation valent respectivment $7{,}5.10^{-9}$ et $8{,}6.10^{-9}$.

VI.5 **Problème n°4** (voir annexe A1)

VI.5.1 *1ère forme du modèle* :

Après réduction du nombre de contraintes en équation (voir V.3.1), ce problème comporte :

- 3 variables x_3, x_4, x_6
- 7 contraintes non linéaires, dont une en équation
- 6 contraintes de bornes (2 sur chaque variable).

La fonction économique est continue et à gradient non continu.

La contrainte en équation est transformée en inéquation par mise en $\leq$.

Le point de départ est le point non réalisable fourni par l'auteur.

1) Essais sans algorithme de centrage

Paramètre de programmation :

lin = 3 dic = 10^4

Valeur de la fonction économique en fonction du numéro de troncature

Troncature	k=1	$k=10^{-2}$
0	100000	100000
2	8862,83	8840,46
4	8844,72	8833,20
6	8840,04	8831,34
8	8837,61	8830,26
10	8836,09	8829,42
15	8833,79	8828,63
20	8832,52	8828,32
25	8831,69	8828,16
30		8828,06

La convergence est très lente, bien que la pondération $k=10^{-2}$ l'accèlère.

Cette lenteur est principalement due à des oscillations sur le point (voir A3.4.1).

2) Essais avec algorithme de centrage (voir I.5)

Paramètres de programmation :

lin = 1 dic = 10^4 $k = 10^{-2}$

Centrage à partir de la 5ème troncature.(choix n°1 de I.5.3)

Valeur de la fonction économique en fonction du numéro de troncature :

0	9519	8	8828,87
1	8856	9	8828,24
2	8842,4	10	8827,92
3	8841,2	12	8827,76
4	8840,65	14	8827,68
5	8838	16	8827,63
6	8832	18	8827,61
7	8829,9	20	8827,60

L'optimum est atteint, mais on a encore quelques ennuis dus à des oscillations sur le point.

3) Meilleure solution obtenue

Le meilleur essai est l'essai avec centrage et avec pour paramètres de programmation lin = 1, dic = 10^4, k = 10^{-2}.

Dans ces conditions :

La valeur 8827,598 (arrondi aux 3 premières décimales) est obtenue à la 25 ème troncature (temps de calcul : 41,2 s).

La meilleure valeur obtenue est

8827,597938

à la 29 ème troncature, ce qui correspond à la solution

x_3 = 373,83106

x_4 = 419,99983

x_6 = 0,1532879

En ce point la contrainte en équation vaut $3,8.10^{-7}$.

VI.5.2 *2ème forme du modèle*

Le nombre de contraintes en équation n'est pas réduit et les variables x_1, x_2, x_5 ne sont pas éliminées. De plus, la fonction économique est totalement linéaire (voir V.3.2)

Sous cette forme, le problème comporte :

- 9 variables $x_{11}, x_{12}, x_{21}, x_{22}, x_{23}, x_3, x_4, x_5, x_6$
- 4 contraintes non linéaires en équation
- 18 contraintes de bornes (2 sur chaque variable).

Les contraintes en équation sont transformées en inéquations par mise en $\leqslant$.

Le point de départ est le point non réalisable fourni par l'auteur.

1) Essais

Paramètre de programmation :

lin = 1 dic = 10^4

Algorithme de centrage (choix n°2 de I.5.3) : 4 contraintes additionnelles à partir de la troncature 1.

Valeurs de la fonction économique en fonction du numéro de troncature :

Troncature	$k=10^{-2}$	$k=10^{-3}$	$k=10^{-4}$	$k=10^{-5}$
0	41490	41490	41490	41490
1	29338,22	29338,22	29338,22	29338,22
2	25147,31	19323,40	11166,43	9941,30
4	22324,37	11576,63	8856,78	8833,25
6	20027,09	9552,24	8828,26	8829,57
8	18144,15	9018,45	8827,63	8827,61
10	16587,22	8881,56	8827,60	8827,61
12	15287,58	8843,46	8827,60	8827,61

Le meilleur résultat est obtenu pour $k=10^{-4}$. Dans ce cas, l'optimum est atteint.

2) Meilleure solution obtenue

Le meilleur essai est l'essai avec $k=10^{-4}$.

Dans ces conditions :

La valeur 8827,598 (arrondi aux 3 premières décimales) est obtenue à la 11ème troncature.

La meilleure valeur obtenue est

8827,59812356

à la 13ème troncature, ce qui correspond à la solution

$x_1 = 107,79178$ $x_3 = 373,82824$

$x_2 = 196,33947$ $x_4 = 420$

$x_5 = 21,31652$ $x_6 = 0,1533099$

En ce point les 4 contraintes en équation valent respectivement 12.10^{-6}, 9.10^{-7}, 22.10^{-7}, 9.10^{-9}. (voir pour cet essai les valeurs numériques en A3.4.2).

VI.6 **Problème n°5** (voir annexe A1)

Pour la résolution par la méthode simpliciale en variables bornées, nous ajoutons arbitrairement des bornes sur les θ_i :

$$-0,5 \leqslant \theta_i \leqslant 0,5 \qquad i = 1,...9.$$

VI.6.1 *1ère forme du modèle :*

Après réduction du nombre de contraintes en équation (voir V.3.1), ce problème comporte :

- 19 variables U_i et θ_i
- 34 contraintes non linéaires, dont 6 en équation
- 38 contraintes de bornes (2 sur chaque variable).

La fonction économique est continue et à gradient non continu.

Les contraintes en équation sont transformées en inéquations par mise en $\leqslant$.

Le point de départ est le point obtenu en mettant les variables à leur borne inférieure.

1) Essais sans algorithme de centrage

Paramètres de programmation :

lin = 3 dic = 10^4 k = 10^{-2}.

Valeur de la fonction économique en fonction du numéro de troncature :

0	7500	16	1546,59
2	1652,83	18	1546,57
4	1578,82	20	1546,55
6	1553,68	22	1546,54
8	1548,52	24	1546,52
10	1546,90	26	1546,51
12	1546,66	28	1546,49
14	1546,60	30	1546,48

La convergence est très lente ; nous avons même une quasi-stagnation autour de 1546,5.

2) Essais avec algorithme de centrage (voir I.5)

Paramètres de programmation :

lin = 1 dic = 10^2 k = 10^{-2}.

4 contraintes de centrage. (choix n°1 de I.5.3)

Valeur de la fonction économique en fonction du numéro de troncature :

1	1511,92	12	1505,34
2	1508,24	14	1505,21
3	1507,75	16	1505,11
4	1507,60	18	1505,04
6	1506,23	20	1505,02
8	1505,77	22	1505,01
10	1505,50	24	1505,01

Ici encore la convergence est lente, et on a une quasi-stagnation autour de 1505.

IV.6.2 *2ème forme du modèle*

Le nombre de contraintes en équation n'est pas réduit.

La fonction économique est linéaire par morceaux ; elle est donc continue et à gradient non continu.

Les variables du problème sont :

- les U_i $\quad i = 1,...10$
- les θ_i $\quad i = 1,...9$
- les P_i $\quad i = 1,2,4,6,8,9,10$
- les Q_i $\quad i = 1,2,4,6,8,9,10.$

Sous cette forme, le problème comporte donc :

- 33 variables
- 20 contraintes non linéaires en équation
- 66 contraintes de bornes (2 sur chaque variable).

Les contraintes en équation sont transformées en inéquations par mise en $\leqslant$.

Le point de départ est le point obtenu en mettant les variables à leur borne inférieure.

1) Essais sans algorithme de centrage

Paramètres de programmation :

lin = 1 dic = 10^2 k = 10^{-2}

Valeur de la fonction économique en fonction du numéro de troncature :

0	7500	18	4402,97
1	4567,71	22	4392,80
2	4554,00	26	4380,57
4	4527,05	30	4370,77
6	4497,91	34	4367,73
8	4478,47	38	4366,98
10	4467,39	42	4366,87
14	4433,26	46.	4366,85

La convergence est très lente.

2) Influence du coefficient k

Paramètres de programmation :

lin = 1 dic = 10^4.

3 contraintes additionnelles. (choix n°1 de I.5.3)

Valeur de la fonction économique en fonction du numéro du troncature :

Troncature	$k=10^{-1}$	$k=10^{-2}$	$k=10^{-4}$	$k=10^{-5}$	$k=10^{-6}$	$k=10^{-7}$
1	4569,9	4567,7	4569,9	4569,9	4569,9	4569,9
2	4568,1	4550,3	3589,1	2488,8	2288,5	2193,4
3	4566,5	4536,4	3065,1	2044,6	2080,6	2141,6
4	4564,7	4519,5	2778,7	1819,7	1970,1	2125,5
5	4563,1	4502,3	2606,0	1725,7	1792,9	2113,6
7	4559,6	4477,2	2300,3	1641,3	1738,8	2090,8
9	4556,1	4456,0	2121,1	1591,0	1715,9	2067,0
11	4552,9	4441,6	1989,0	1571,2	1694,8	2049,3
13	4549,4	4417,9	1915,5	1558,4	1656,7	2028,7
15	4545,9	4397,1	1842,6	1548,9	1643,9	2005,9

Les meilleurs résultats sont obtenus avec $k=10^{-5}$.

3) Influence du nombre de contraintes de centrage (voir I.5)

Paramètres de programmation :

lin = 1 dic = 10^4 k = 10^{-5}

Valeur de la fonction économique en fonction du numéro de troncature, pour des nombres différents de contraintes additionnelles (choix n°1 de I.5.3).

Troncature	2contraintes	3contraintes	9contraintes
1	4569,9	4569,9	4569,9
2	2488,8	2488,8	2304,6
4	1877,2	1819,7	1702,1
6	1720,3	1657,8	1581,6
8	1626,4	1616,4	1556,8
10	1602,0	1586,0	1539,7
14	1579,1	1551,2	1517,9
18	1572,1	1538,4	1512,1
22	1557,6	1533,8	1505,0

L'augmentation du nombre de contraintes additionnelles accélère la convergence ; cependant on a des difficultés à arriver à l'optimum dues à des oscillations sur le point (voir A3.5.1) en particulier sur les P_i qui oscillent autour de leur valeur optimale, comme le montre la courbe (figure VI.5)

VI.6.3 *3ème forme du modèle*

le nombre de contraintes en équation n'est pas réduit.

La fonction économique est entièrement linéaire (voir V.3.2)

Les variables du problème sont :

- les U_i. $\quad i = 1,\ldots 10$
- les θ_i $\quad i = 1,\ldots 9$
- les P_{ij} $\quad i = 1,2,4,6,8,9,10 \quad j \in J_i$
- les Q_i. $\quad i = 1,2,4,6,8,9,10.$

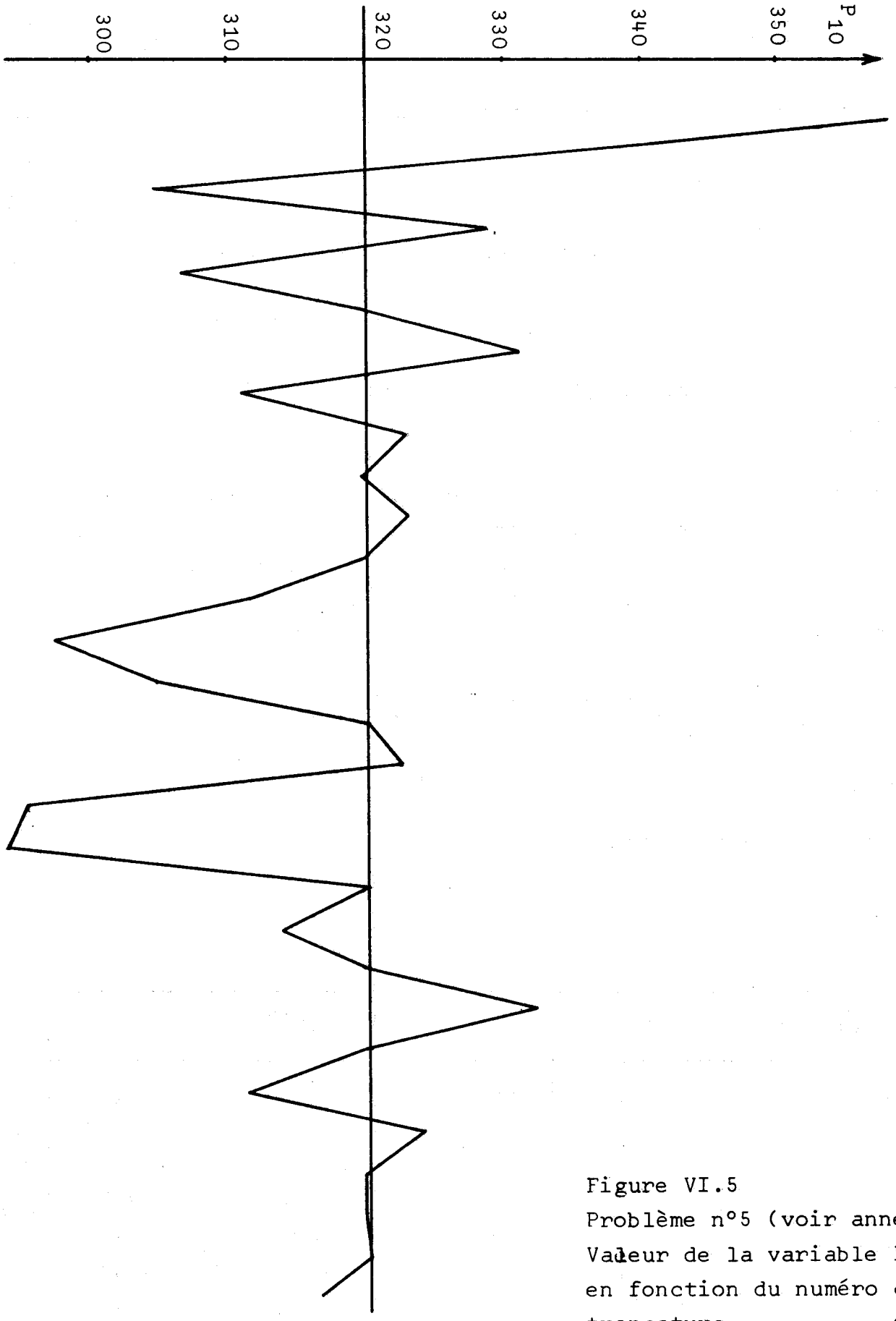


Figure VI.5
Problème n°5 (voir annexe A1)
Valeur de la variable P_{10}
en fonction du numéro de
troncature

Sous cette forme, le problème comporte donc :

- 50 variables

- 20 contraintes non linéaires en équation

-100 contraintes de bornes (2 sur chaque variable).

Les contraintes en équation sont transformées en inéquations par mise en $\leqslant$.

Dans les essais suivants nous utiliserons 2 points de départ : le point P_1 obtenu en mettant les variables à leur borne inférieure, et le point P_2, plus proche de l'optimum , obtenu au cours d'un essai (et donc réalisable), correspondant à une fonction économique de 1500 environ.

1) Influence du coefficient k

1er essai : sans algorithme de centrage.

Paramètres de programmation :

lin = 1 dic = 10^4 départ du point P_1

Troncature	$k=10^{-2}$	$k=10^{-4}$	$k=10^{-5}$	$k=10^{-6}$
1	4569,9	4569,9	4569,9	4569,9
2	4546,3	3354,7	2457,1	2297,1
3	4521,9	2479,2	2206,0	2211,9
4	4500,4	2136,3	1651,7	2163,9
6	4450,8	1855,3	1557,8	2113,5
8	4410,5	1749,7	1531,8	1697,0
10	4361,6	1610,3	1525,8	1565,9

2ème essai :

Algorithme de centrage, départ de P_1.

Paramètres de programmation :

lin = 1 dic = 10^4 4 contraintes de centrage (choix n°1 de I.5.3)

Troncature	$k=10^{-4}$	$k=10^{-5}$
1	4569,9	4569,9
2	3089,0	2017,9
3	2352,7	1661,6
4	1965,5	1580,6
6	1672,2	1542,1
8	1560,1	1521,2
10	1527,2	1508,6

3ème essai :

algorithme de centrage, départ de P_2.

Paramètres de programmation :

lin = 1 dic = 10^9 4 contraintes de centrage (choix n°1 de I.5.3)

Troncature	$k=10^{-3}$	$k=10^{-4}$	$k=10^{-5}$	$k=10^{-6}$
1	1499,964	1499,964	1499,964	1499,964
2	1499,946	1499,904	1499,880	1499,909
3	1499,918	1499,799	1499,768	1499,895
4	1499,903	1499,766	1499,681	1499,861
6	1499,867	1499,691	1499,550	1499,813
8	1499,825	1499,609	1499,504	1499,773

Dans tous les cas, les meilleurs résultats sont obtenus avec $k = 10^{-5}$.

2) Influence du nombre de contraintes de centrage

1er essai :

départ de P_1.

Paramètres de programmation :

lin = 1 dic = 10^4 k = 10^{-5}

centrage (choix n°1 de I.5.3)

Troncature	4 contraintes de centrage	5 contraintes de centrage
1	4569,9	4569,9
2	2017,9	2017,9
4	1580,6	1545,2
6	1542,1	1518,4
8	1521,2	1513,2
10	1508,6	1506,3
12	1501,9	1502,3
14	1500,3	1501,0

Ici l'addition d'un contrainte de centrage par troncature (passage de 4 à 5) n'a pas accéléré la convergence.

2ème essai :

départ de P_2.

Paramètres de programmation :

lin = 1 dic = 10^9 k = 10^{-5}

centrage (choix n°1 de I.5.3)

Troncature	4 contraintes de centrage	6 contraintes de centrage
1	1499,964	1499,964
2	1499,880	1499,848
4	1499,681	1499,626
6	1499,550	1499,427
8	1499,504	1499,390

3ème essai :

départ de P_2,

Paramètre de programmation :

lin = 1 dic = 10^9 k = 10^{-5}

Centrage (choix n°2 de I.5.3)

Troncature	8 contraintes de centrage	12 contraintes de centrage
1	1499,964	1499,964
2	1499,776	1499,652
3	1499,390	1499,392
4	1499,337	1499,264
5	1499,263	1499,241
6	1499,254	1499,237

Nous voyons qu'une augmentation importante du nombre de contraintes de centrage (passage de 8 à 12) a eu un effet sensible sur la rapidité de convergence.

3) Influence de la forme du centrage

Paramètres de programmation :

lin = 1 dic = 10^9 k = 10^{-5}

Départ du point P_2 8 contraintes de centrage

2 essais sont comparés, l'un avec le choix N°1, l'autre avec le choix N°2 de I.5.3

Troncature	Choix N°1	Choix N°2
1	1499,964	1499,964
2	1499,848	1499,776
3	1499,706	1499,390
4	1499,626	1499,337
5	1499,548	1499,263
6	1499,427	1499,254

Avec le choix n°2 la convergence est plus rapide.

4) Meilleure solution obtenue

La valeur 1499,24 (arrondi aux 2 premières décimales) est obtenue à la 13ème troncature, avec les paramètres de programmation lin = 1, dic = 10^9, k = 10^{-5}, et 12 contraintes de centrage (choix N°2 de I.5.3) à partir de la 8ème troncature, comme indiqué sur la figure VI.6 (voir les valeurs numériques correspondantes en A3:5.2) Temps de calcul 2856 s avec la dichotomie (III.2.2) et 1692 s avec l'interpolation polygonale (III.3.2)

La meilleure valeur obtenue est

1499,23582977

obtenue à la 11ème troncature à partir du point P_2, avec les paramètres de programmation lin = 1, dic 10^9, k = 10^{-5} et 12 contraintes de centrage (choix n°2 de I.5.3) , comme indiqué sur la figure VI.7 (voir les valeurs numériques correspondantes en A3.5.3).

Cette valeur correspond à la solution

noeud i	P_i	Q_i	U_i	θ_i
1	256,411	108,420	240	- 0,006638
2	222,674	7,290	240	0,017146
3	0	0	221,756	- 0,112668
4	560	390	233,518	- 0,053546
5	0	0	231,879	- 0,133939
6	140	112,296	239,287	- 0,083267
7	0	0	233,417	- 0,065269
8	320	149,058	239,430	- 0,021354
9	150	55,329	240	- 0,103094
10	320	88,175	240	0

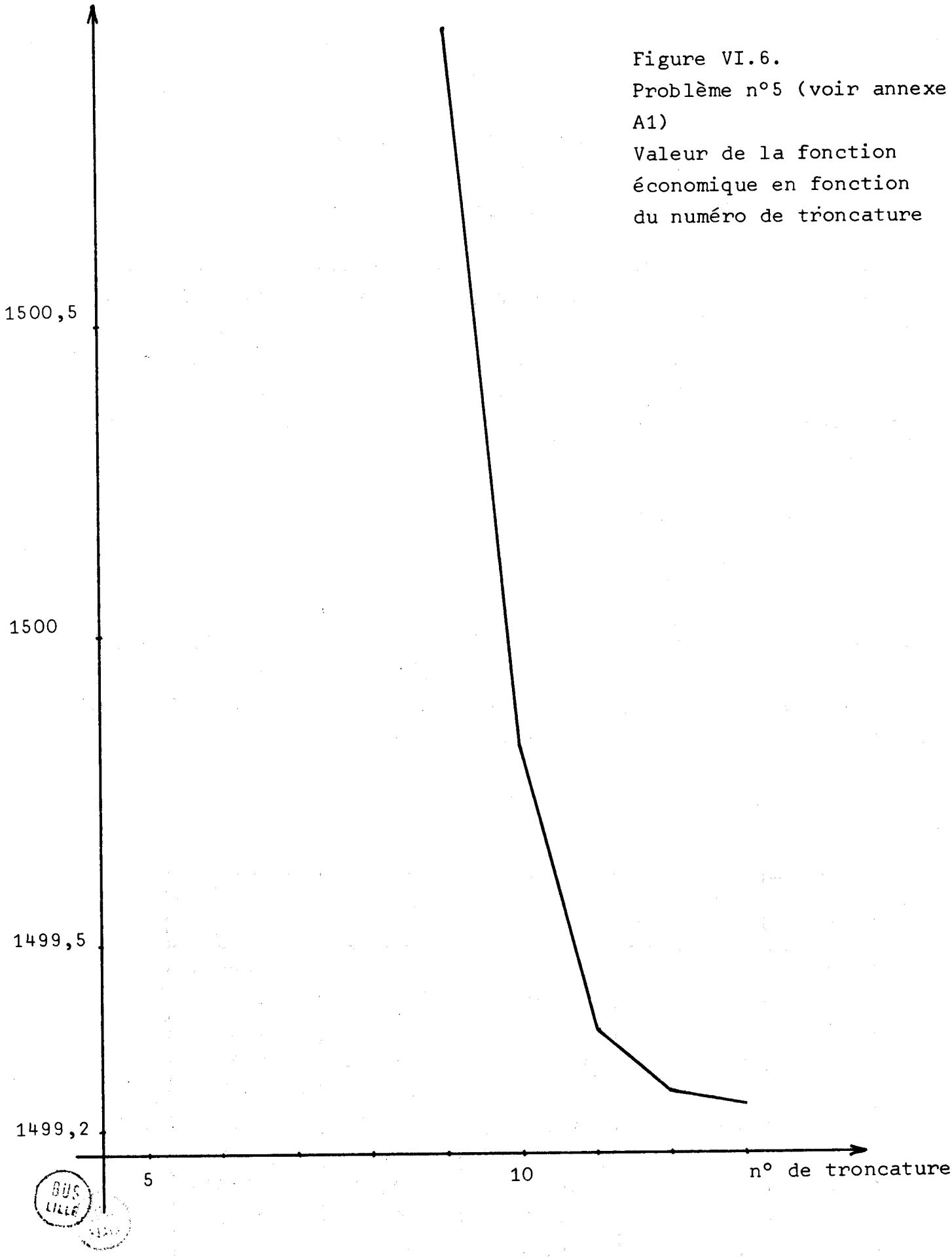


Figure VI.6.
Problème n°5 (voir annexe A1)
Valeur de la fonction économique en fonction du numéro de troncature

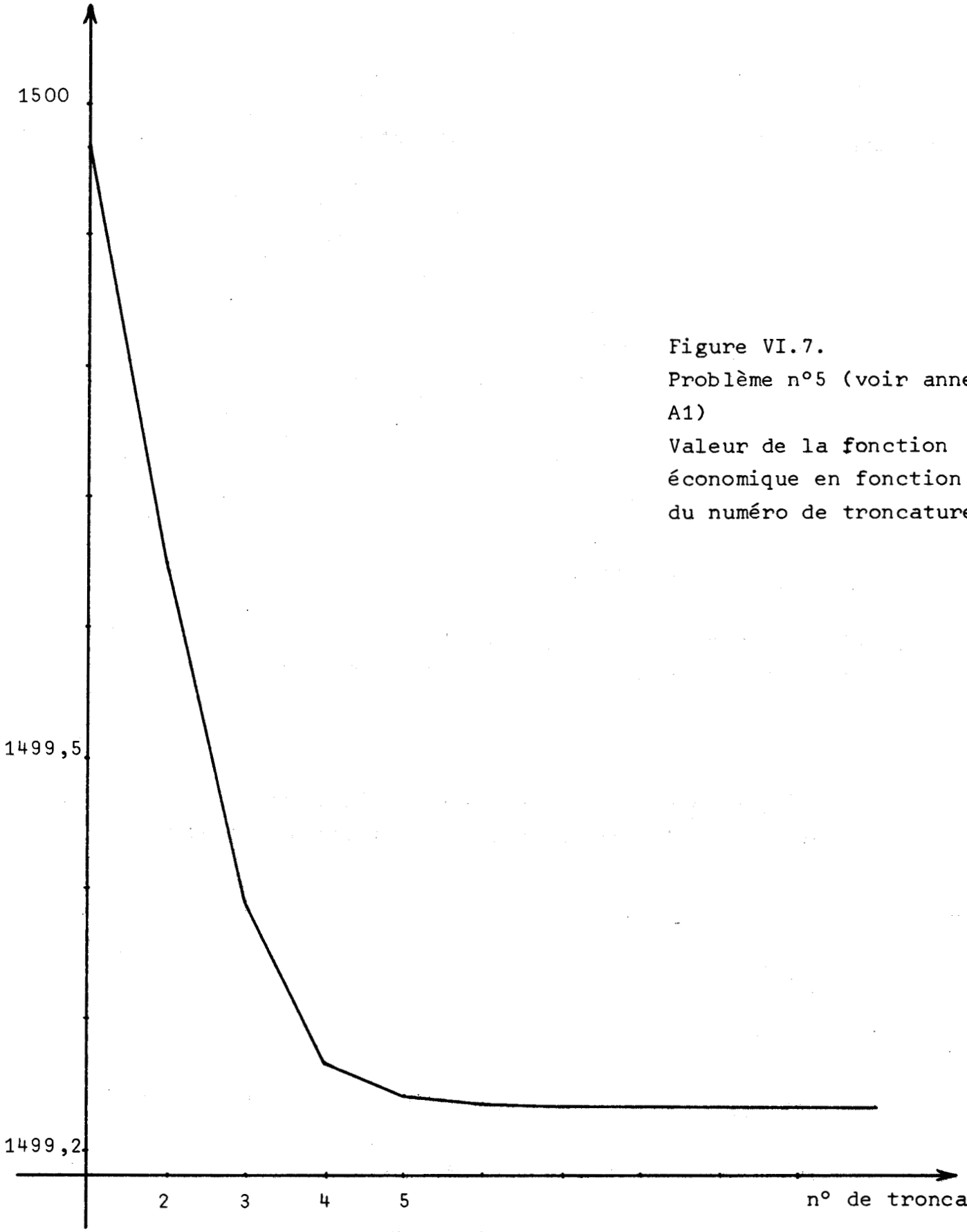


Figure VI.7.
Problème n°5 (voir annexe A1)
Valeur de la fonction économique en fonction du numéro de troncature

En ce point les 20 contraintes en équation valent :

$9,7.10^{-8}$	0
$2,5.10^{-6}$	0
$5,1.10^{-6}$	$1,1.10^{-6}$
$6,2.10^{-6}$	0
$3,1.10^{-7}$	$3,5.10^{-6}$
$4,1.10^{-5}$	0
$4,3.10^{-6}$	$5,2.10^{-6}$
$5,8.10^{-5}$	0
$1,1.10^{-5}$	0
$3,8.10^{-5}$	0

VI.7 **Problème n°6** (voir annexe A1).

Pour la résolution par la méthode simpliciale en variables bornées, nous ajoutons arbitrairement des bornes sur les θ_i :

$$-1 \leqslant \theta_i \leqslant 1 \qquad i = 1,...43.$$

Dans le modèle considéré, le nombre de contraintes en équation n'est pas réduit et la fonction économique est entièrement linéaire

Les variables du problème sont :

- les U_i $\qquad i = 1,....44$
- les θ_i $\qquad i = 1,....43$
- les P_{ij} $\qquad i = 11,24,25,29,44 \qquad j \in J_i$
- les Q_i $\qquad i = 11,24,25,29,44$

Sous cette forme, le problème comporte donc :

- 103 variables

- 88 contraintes non linéaires en équation

- 206 contraintes de bornes (2 sur chaque variable).

Les contraintes du réactif correspondant aux noeuds 6,7,8,9,10,15,16, 17,18,19,20,21,22,23,28 sont transformées en inéquations par mise en $\geqslant$. Toutes les autres contraintes sont mises en $\leqslant$ (voir IV.6)

Le point de départ est le point obtenu en mettant les puissances P_i et Q_i à leur borne supérieure et les autres variables à leur borne inférieure. L'algorithme de I.6.2, avec la fonction économique non pondérée, fournit un point réalisable au bout de 6 linéarisations.

1) Ajustement du coefficient k.

Paramètres de programmation :

lin = 1 dic = 10^9 pas de centrage

Troncature	$k=10^{-4}$	$k=10^{-5}$	$k=10^{-6}$
1	84,442	84,442	84,442
2	82,901	78,677	76,613
3	78,901	72,786	74,379
4	76,952	66,720	72,632
5	74,743	63,349	71,052
6	72,790	60,839	70,004
7	71,099	58,634	68,703
8	69,314	56,787	67,236

Ces valeurs sont représentées sur la ligne VI.8.

Le meilleur résultat est obtenu avec $k = 10^{-5}$. La convergence reste cependant encore lente.

2) Essais avec algorithme de centrage

Paramètres de programmation :

lin = 1 dic = 10^9 $k = 10^{-5}$

centrage (choix N°2 de I.5.3)

Troncature	5 contraintes de centrage	10 contraintes de centrage	15 contraintes de centrage
1	84,442	84,442	84,442
2	71,307	68,242	68,242
3	60,048	57,848	57,848
4	53,317	51,334	50,628
5	49,383	47,277	46,514
6	46,541	44,999	44,499
8	43,810	43,286	43,157
10	43,101	42,960	42,935
12	42,937	42,908	42,903

Ces valeurs sont représentées sur la figure VI.9
L'adjonction de contraintes de centrage accélère nettement la convergence.

Figure VI.8.
Problème n°6 (voir annexe A1)
Valeur de la fonction économique en fonction du numéro de troncature

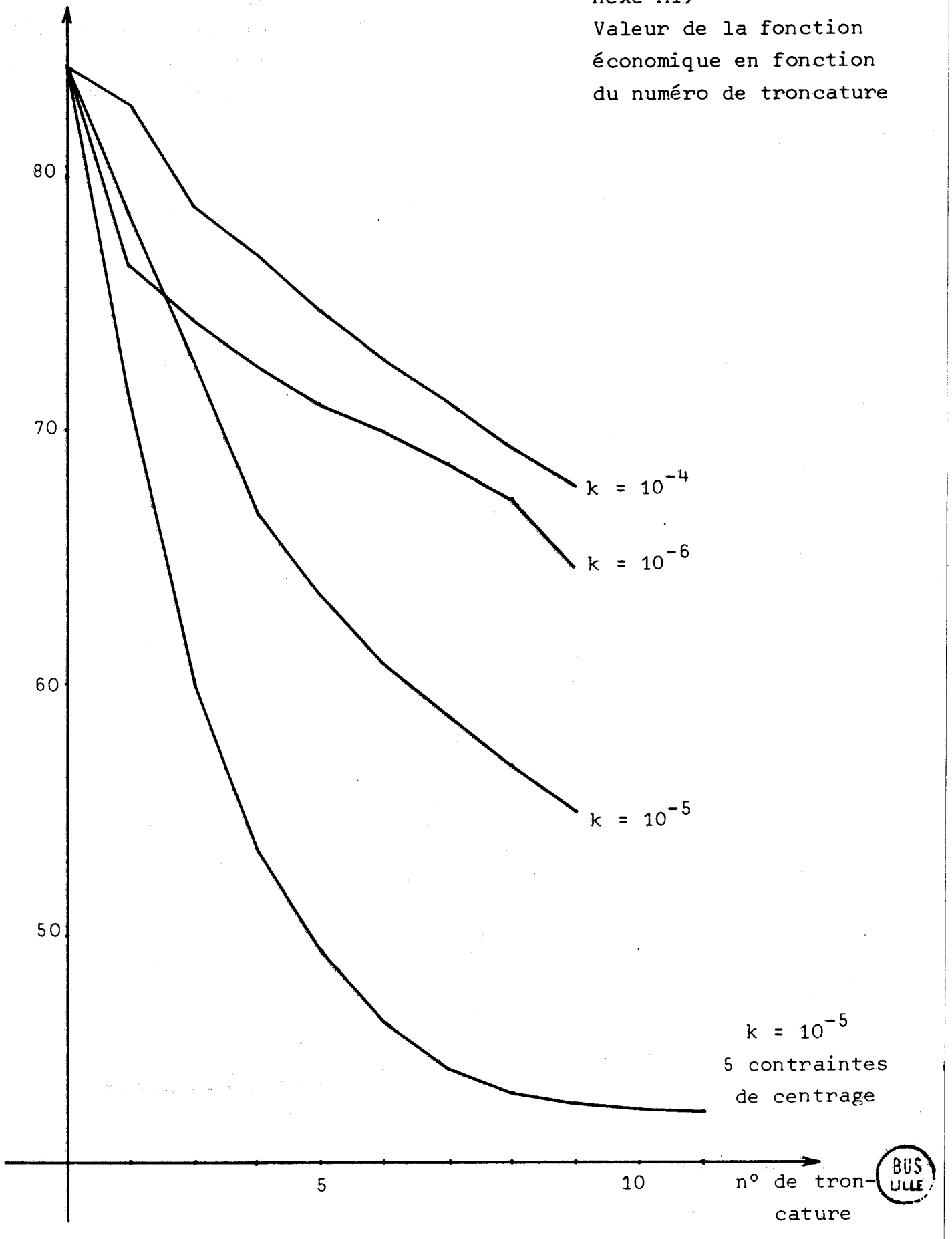
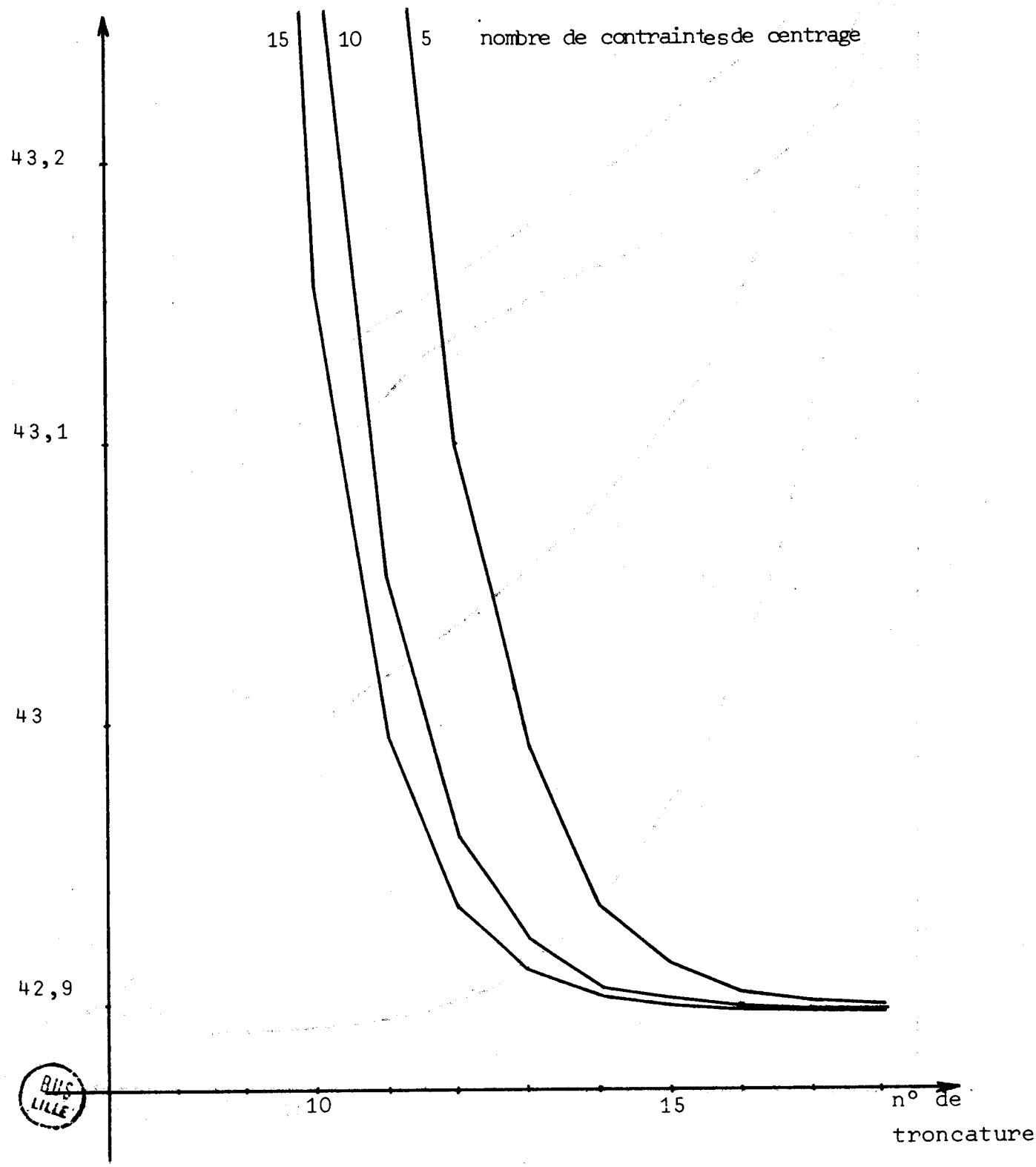


Figure VI.9
Problème n°6(voir annexe A1)
valeur de la fonction économique en fonction du numéro de troncature



3) Meilleure solution obtenue.

La valeur 42,898 (arrondi aux 3 premières décimales) est obtenue à la 15ème troncature, avec les paramètres de programmation lin =1, dic = 10^9, k = 10^{-5}, et 15 contraintes de centrage (choix N°2 de I.5.3) dès la 1ère troncature.
Temps de calcul : 5495 s.

La valeur 42,8982 (arrondi aux 4 premières décimales) est obtenue à la 17ème troncature, avec les mêmes paramètres.
Temps de calcul : 6108 s.

La meilleure valeur obtenue est

42,8981784084

obtenue à la 19ème troncature avec les paramètres de programmation précédents (voir les valeurs numériques en A3.6)

Solution :

Noeud i	P_i	Q_i	U_i	θ_i
0	332,0000	72,6484	168,0000	
1			152,6805	-0,277133
2			162,0644	-0,068160
3			152,0908	-0,211713
4			150,2078	-0,218637
5			152,4286	-0,208466
6			168,0000	-0,118419
7			167,9444	-0,116324
8			166,7366	-0,121455
9			157,5234	-0,288015
10			157,7756	-0,288452
11	10,0000	25,8614	149,1490	-0,505063
12			145,6534	-0,505885
13			143,1034	-0,553907
14			146,0305	-0,449642
15			151,7700	-0,344045
16			155,3721	-0,307873
17			157,6798	-0,285795
18			161,6918	-0,237917
19			160,3494	-0,240868
20			154,4858	-0,284463
21			157,8497	-0,241211
22			166,1695	-0,122298
23			148,7437	-0,376062
24	0,0000	-0,9622	141,6358	-0,487284
25	125,2010	79,2610	144,2164	-0,522038
26			145,2408	-0,480428
27			145,6214	-0,458665
28			167,2874	-0,068573
29	23,0000	20,0000	143,5818	-0,544043
30			145,0159	-0,503260
31			148,8207	-0,346574
32			164,5481	-0,022777
33			164,8156	-0,024955
34			155,7085	-0,080193
35			155,7577	-0,078873

36			153,0563	-0,123220
37			153,1252	-0,120846
38			153,6653	-0,114184
39			154,5639	-0,100256
40			155,6708	-0,076392
41			155,7519	-0,074754
42			156,0586	-0,070630
43			162,5622	-0,036031

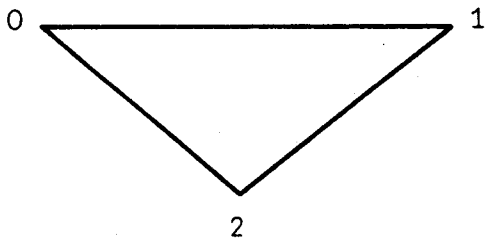
ANNEXES

A N N E X E 1

Exemples de réseaux

A.1.1 PROBLEME N°1

Réseau à 3 noeuds, actif seul, avec limites de transit ([6] page 69).



Minimiser $(3P_0 + 10^{-6}P_0^3 + 2P_1 + \frac{2}{3}\,10^{-6}P_1^3)$

sous les contraintes

$$1000 \sin(-\theta_1 - 0{,}25) + 1000 \sin(-\theta_2 - 0{,}25) + 894{,}8 = P_0$$

$$1000 \sin(\theta_1 - 0{,}25) + 1000 \sin(\theta_1 - \theta_2 - 0{,}25) + 894{,}8 = P_1$$

$$1000 \sin(\theta_2 - 0{,}25) + 1000 \sin(\theta_2 - \theta_1 - 0{,}25) + 1294{,}8 = 0$$

$$0 \leq P_0 \leq 1200$$
$$0 \leq P_1 \leq 1200$$
$$-0{,}55 \leq \theta_1 \leq 0{,}55 \qquad \theta_1 - \theta_2 \leq 0{,}55$$
$$-0{,}55 \leq \theta_2 \leq 0{,}55 \qquad \theta_2 - \theta_1 \leq 0{,}55$$

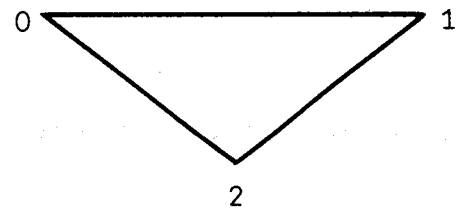
Solution donnée dans [6] :

Optimum : 5125,2

Solution : $\theta_1 = 0{,}119$ $\quad P_0 = 679{,}6$
$\theta_2 = -0{,}396$ $\quad P_1 = 1026{,}1$

A1.2 PROBLEME N°2

Réseau à 3 noeuds, actif seul, avec limites de transit ([6] page 69).



Seule la partie linéaire du problème précédent est modifiée.

Minimiser $(3P_0 + 10^{-6}P_0^3 + 2P_1 + \frac{2}{3}\,10^{-6}P_1^3)$

Sous les contraintes

$$1000 \sin(-\theta_1 - 0{,}25) + 1000\sin(-\theta_2 - 0{,}25) + 894{,}8 = P_0$$

$$1000 \sin(\theta_1 - 0{,}25) + 1000\sin(\theta_1 - \theta_2 - 0{,}25) + 894{,}8 = P_1$$

$$1000 \sin(\theta_2 - 0{,}25) + 1000\sin(\theta_2 - \theta_1 - 0{,}25) + 1294{,}8 = 0$$

$$0 \leq P_0 \leq 1200$$

$$0 \leq P_1 \leq 1200$$

$$-0{,}48 \leq \theta_1 \leq 0{,}48 \qquad \theta_1 - \theta_2 \leq 0{,}48$$

$$-0{,}48 \leq \theta_2 \leq 0{,}48 \qquad \theta_2 - \theta_1 \leq 0{,}48$$

Solution donnée dans [6]:

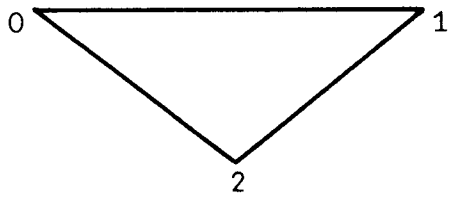
Optimum : 5158,6

Solution : $\theta_1 = 0{,}054$ $\quad P_0 = 770{,}6$

$\theta_2 = -0{,}426$ $\quad P_1 = 928{,}1$

A.1.3 PROBLEME N°3

Réseau à 3 noeuds, actif et réactif, avec limite de transit ([6]page 62).



Minimiser $(3P_0 + 10^{-6}P_0^3 + 2P_1 + 0{,}522074.10^{-6}P_1^3)$

sous les contraintes

$$\frac{U_0U_1}{50{,}176}\sin(-\theta_1 - 0{,}25) + \frac{U_0U_2}{50{,}176}\sin(-\theta_2 - 0{,}25) + 2\frac{U_0^2 \sin 0{,}25}{50{,}176} + 400 = P_0$$

$$\frac{U_1U_0}{50{,}176}\sin(\theta_1 - 0{,}25) + \frac{U_1U_2}{50{,}176}\sin(\theta_1 - \theta_2 - 0{,}25) + 2\frac{U_1^2 \sin 0{,}25}{50{,}176} + 400 = P_1$$

$$\frac{U_2U_0}{50{,}176}\sin(\theta_2 - 0{,}25) + \frac{U_2U_1}{50{,}176}\sin(\theta_2 - \theta_1 - 0{,}25) + 2\frac{U_2^2 \sin 0{,}25}{50{,}176} + 881{,}779 = 0$$

$$-\frac{U_0U_1}{50{,}176}\cos(-\theta_1 - 0{,}25) - \frac{U_0U_2}{50{,}176}\cos(-\theta_2 - 0{,}25) + 2\frac{U_0^2 \cos 0{,}25}{50{,}176} - 0{,}7533.10^{-3}U_0^2 + 200 = Q_0$$

$$-\frac{U_1U_0}{50{,}176}\cos(\theta_1 - 0{,}25) - \frac{U_1U_0}{50{,}176}\cos(\theta_1 - \theta_2 - 0{,}25) + 2\frac{U_1^2 \cos 0{,}25}{50{,}176} - 0{,}7533.10^{-3}U_1^2 + 200 = Q_1$$

$$-\frac{U_2U_0}{50{,}176}\cos(\theta_2 - 0{,}25) - \frac{U_2U_1}{50{,}176}\cos(\theta_2 - \theta_1 - 0{,}25) + 2\frac{U_2^2 \cos 0{,}25}{50{,}176} - 0{,}7533.10^{-3}U_2^2 + 22{,}938 = 0$$

$0 \leq P_0$ $\qquad -400 \leq Q_0 \leq 800$

$0 \leq P_1$ $\qquad -400 \leq Q_1 \leq 800$

$P_0^2 + Q_0^2 \leq 1500^2$

$P_1^2 + Q_1^2 \leq 1500^2$

$-0,55 \leq \theta_1 \leq 0,55$ $\qquad \theta_1 - \theta_2 \leq 0,55$

$-0,55 \leq \theta_2 \leq 0,55$ $\qquad \theta_2 - \theta_1 \leq 0,55$

$196 \leq U_i \leq 252$ $\qquad i = 0,1,2.$

Solution donnée dans [6]

Optimum : 5362,903

Solution : $U_0 = 252$ $\qquad P_0 = 688,902$
$U_1 = 252$ $\qquad P_1 = 1118,934$
$U_2 = 201,6$ $\qquad Q_0 = 422,417$
$\theta_1 = 0,125$ $\qquad Q_1 = 367,668$
$\theta_2 = -0,375$

A.1.4. PROBLEME N°4

Réseau à 2 noeuds, actif et réactif, avec limites de transit (problème n°6 de [8]).

Minimiser $[f_1(x_1) + f_2(x_2)]$

$$\frac{df_1}{dx_1} = \begin{cases} 30 \text{ si } 0 \leq x_1 \leq 300 \\ 31 \text{ si } 300 \leq x_1 \leq 400 \end{cases} \qquad \frac{df_2}{dx_2} = \begin{cases} 28 \text{ si } 0 \leq x_2 \leq 100 \\ 29 \text{ si } 100 \leq x_2 \leq 200 \\ 30 \text{ si } 200 \leq x_2 \leq 1000 \end{cases}$$

sous les contraintes

$$C - \frac{x_3 x_4}{B} \cos (b - x_6) + x_3^2 \frac{A}{B} \cos (b - a) = x_1$$

$$- \frac{x_3 x_4}{B} \cos (b + x_6) + x_4^2 \frac{A}{B} \cos (b - a) = x_2$$

$$D - \frac{x_3 x_4}{B} \sin (b - x_6) + x_3^2 \frac{A}{B} \sin (b - a) = 0$$

$$- \frac{x_3 x_4}{B} \sin (b + x_6) + x_4^2 \frac{A}{B} \sin (b - a) = x_5$$

$A = 0,90798$ $a = 0,00889$ $C = 300$

$B = 131,078$ $b = 1,48477$ $D = 200$

$0 \leqslant x_1 \leqslant 400$ $340 \leqslant x_3 \leqslant 420$

$0 \leqslant x_2 \leqslant 1000$ $340 \leqslant x_4 \leqslant 420$

$-1000 \leqslant x_5 \leqslant 1000$ $0 \leqslant x_6 \leqslant 0,5236$

Point non réalisable de départ :

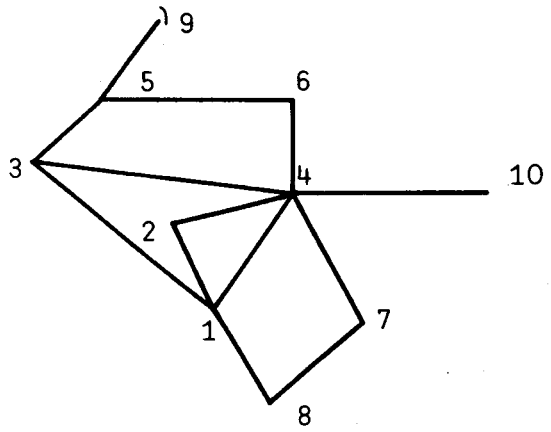
$x_1 = 390,0$ $x_3 = 419,5$

$x_2 = 1000,0$ $x_4 = 340,5$

$x_5 = 191,175$ $x_6 = 0,50$

A.1.5. PROBLEME N°5

Réseau à 10 noeuds, actif et réactif, sans limites de transit (CIGRE).



Minimiser $(1{,}87\ P_{1,1} + 1{,}92\ P_{1,2} + 2{,}3\ P_{1,3} + 2{,}61\ P_{1,4}$

$+ 1{,}85\ P_{2,1} + 1{,}88\ P_{2,2} + 1{,}89\ P_{4,1} + 1{,}90\ P_{4,2} + 2{,}27\ P_{4,3}$

$+ 2{,}28\ P_{4,4} + 2{,}36\ P_{6,1} + 2{,}38\ P_{6,2} + 2{,}51\ P_{6,3} + 2{,}54\ P_{6,4}$

$+ 2{,}56\ P_{6,5} + 1{,}68\ P_{8,1} + 2{,}33\ P_{8,2} + 2{,}43\ P_{8,3} + 1{,}96\ P_{9,1}$

$+ 2{,}18\ P_{9,2} + 1{,}77\ P_{10,1} + 2{,}25\ P_{10,2} + 2{,}31\ P_{10,3} + 2{,}62\ P_{10,4})$

les $P_{i,j}$ étant calculés à partir des P_i comme indiqué ci-dessous sous les contraintes :

$$- \sum_{j \in e(i)} \frac{U_i U_j}{B_{ij}} \cos(\theta_i - \theta_j + \beta_{ij}) + \sum_{j \in e(i)} U_i^2 \frac{A_{ij}}{B_{ij}} \cos(\beta_{ij} - \alpha_{ij}) + C_i = P_i$$

$$- \sum_{j \in e(i)} \frac{U_i U_j}{B_{ij}} \sin(\theta_i - \theta_j + \beta_{ij}) + \sum_{j \in e(i)} U_i^2 \frac{A_{ij}}{B_{ij}} \sin(\beta_{ij} - \alpha_{ij}) + D_i = Q_i$$

$80 \leq P_{1,1} \leq 240$

$80 \leq P_{1,2} \leq 240$

$40 \leq P_{1,3} \leq 120$

$20 \leq P_{1,4} \leq 50$

$40 \leq P_{6,1} \leq 120$

$40 \leq P_{6,2} \leq 120$

$20 \leq P_{6,3} \leq 50$

$20 \leq P_{6,4} \leq 50$

$20 \leq P_{6,5} \leq 50$

$P_3 = 0$

$-113 \leq Q_1 \leq 335$

$-48 \leq Q_2 \leq 240$

$-78 \leq Q_4 \leq 390$

$-45 \leq Q_6 \leq 210$

$Q_3 = 0$

$205 \leq U_i \leq 240 \quad \forall i$

$-0,5 \leq \theta_i \leq 0,5 \quad \forall i \neq 10$

$80 \leq P_{2,1} \leq 240$

$80 \leq P_{2,2} \leq 240$

$80 \leq P_{8,1} \leq 240$

$40 \leq P_{8,2} \leq 120$

$40 \leq P_{8,3} \leq 120$

$40 \leq P_{9,1} \leq 120$

$30 \leq P_{9,2} \leq 80$

$P_5 = 0$

$Q_5 = 0$

$\theta_{10} = 0$

$80 \leq P_{4,1} \leq 240$

$80 \leq P_{4,2} \leq 240$

$40 \leq P_{4,3} \leq 120$

$40 \leq P_{4,4} \leq 120$

$80 \leq P_{10,1} \leq 240$

$40 \leq P_{10,2} \leq 120$

$20 \leq P_{10,3} \leq 50$

$20 \leq P_{10,4} \leq 50$

$P_7 = 0$

$-54 \leq Q_8 \leq 270$

$-23 \leq Q_9 \leq 115$

$-49 \leq Q_{10} \leq 235$

$Q_7 = 0$

i	C_i	D_i	i	C_i	D_i
1	0	0	6	100	35
2	0	0	7	100	50
3	250	160	8	250	150
4	1000	630	9	100	30
5	150	75	10	0	0

Sommets reliés i	j	A_{ij}	$1/B_{ij}$	α_{ij}	β_{ij}
1	2	0,9951	0,0401	0,0010	1,3698
1	3	0,9944	0,0350	0,0012	1,3686
1	4	0,9882	0,0251	0,0018	1,4207
1	8	0,9875	0,0151	0,0046	1,2230
2	4	0,9976	0,0400	0,0005	1,3696
3	4	0,9806	0,0100	0,0050	1,3243
3	5	0,9807	0,0101	0,0050	1,3231
4	6	0,9975	0,0400	0,0004	1,4205
4	7	0,9901	0,0295	0,0025	1,3266
4	10	0,9980	0,0981	0,0004	1,3735
5	6	0,9936	0,0302	0,0019	1,2811
5	9	0,9882	0,0251	0,0018	1,4207
7	8	0,9903	0,0301	0,0025	1,3216

Calcul des $P_{i,j}$ à partir des P_i :

- en chaque noeud où $P_i \neq 0$, calcul de la somme des bornes inférieures des $P_{i,j}$: $\sum_{j \in J_i} P^m_{i,j}$

-déduction de cette somme de P_i :
$P'_i = P_i - \sum_{j \in J_i} P^m_{i,j}$

-ventilation de P'_i sur les différents groupes j pour $j \in J_i$ dans l'ordre des coûts croissants, par saturation successive des groupes.

Solution donnée.

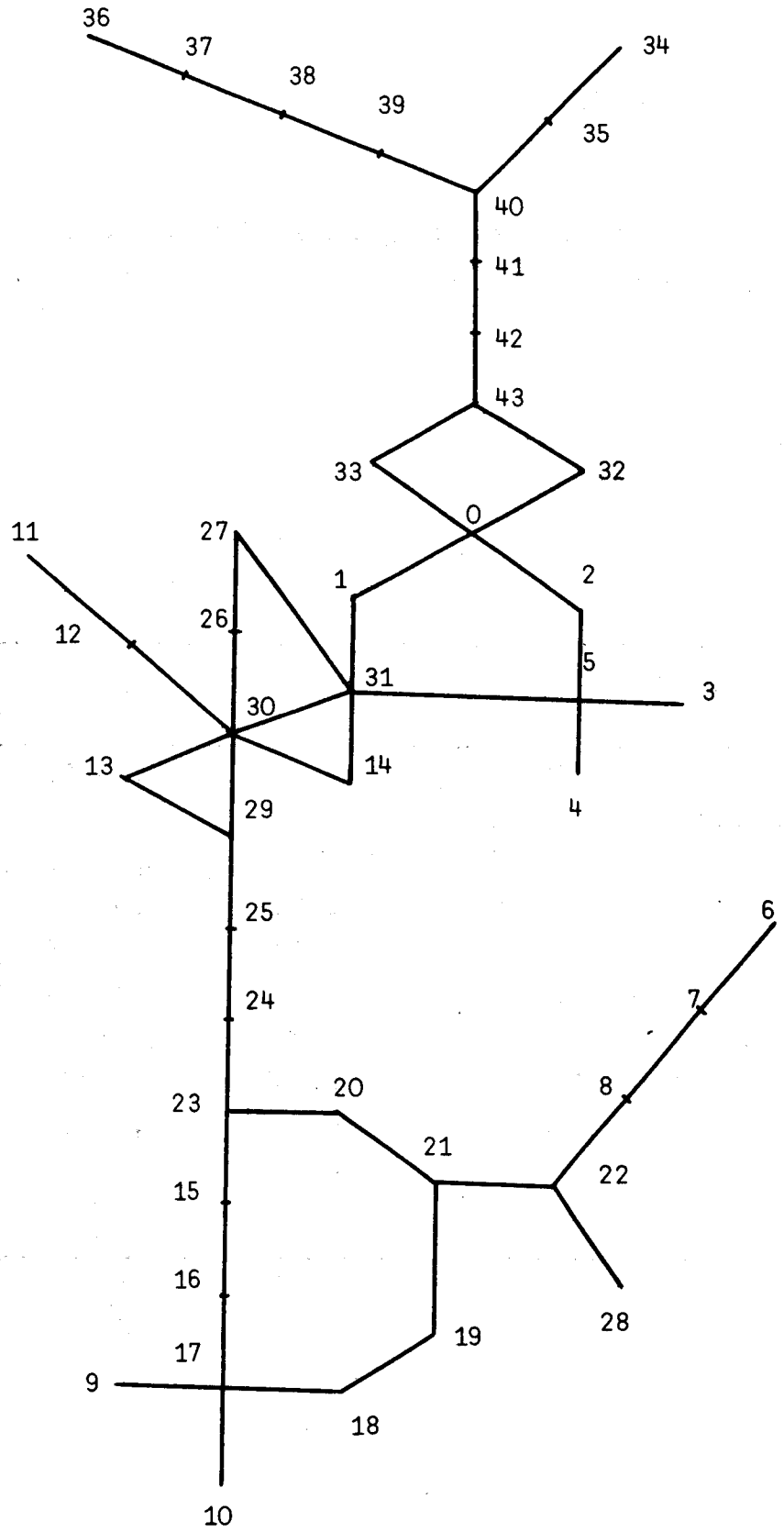
Optimum : 1498,95

Solution :

Noeud i	P_i	Q_i	U_i	θ_i
1	256,31	108,45	240	- 0,00669
2	222,62	7,18	240	0,01711
3	0	0	221,77	- 0,11255
4	560	390	233,52	- 0,05354
5	0	0	231,89	- 0,13396
6	140	112,46	239,30	- 0,08327
7	0	0	233,42	- 0,06528
8	320	149,23	239,44	- 0,02139
9	150	55,23	240	- 0,10316
10	320	88,06	240	0

A.1.6. PROBLEME N°6

Réseau à 44 noeuds, actif et réactif, sans limites de transit (réseau grec).



Minimiser $(0{,}06(P_{0,1} + P_{0,2}) + 0{,}07(P_{0,3} + P_{0,4} + P_{0,5}) + 0{,}08 (P_{0,6} + P_{0,7}) + 0{,}26\ P_{11,1} + 0{,}35\ P_{11,2} + 0{,}36\ P_{24,1} + 0{,}12\ P_{25,1} + 0{,}19\ P_{25,2} + 0{,}15\ P_{29,1} + 0{,}22\ P_{29,2} + 0{,}25\ P_{29,3})$

Les $P_{i,j}$ étant calculés à partir des P_i comme dans le problème n°5

sous les contraintes :

$$- \sum_{j \in e(i)} \frac{U_i U_j}{B_{ij}} \cos(\theta_i - \theta_j + \beta_{ij}) + \sum_{j \in e(i)} U_i^2 \frac{A_{ij}}{B_{ij}} \cos(\beta_{ij} - \alpha_{ij}) + C_i = P_i$$

$$- \sum_{j \in e(i)} \frac{U_i U_j}{B_{ij}} \sin(\theta_i - \theta_j + \beta_{ij}) + \sum_{j \in e(i)} U_i^2 \frac{A_{ij}}{B_{ij}} \sin(\beta_{ij} - \alpha_{ij}) + D_i = Q_i$$

$182 \leq P_{0,1} \leq 215$ $\quad$ $10 \leq P_{11,1} \leq 62$ $\quad$ $3 \leq P_{29,1} \leq 23$

$0 \leq P_{0,2} \leq 23$ $\quad$ $0 \leq P_{11,2} \leq 14$ $\quad$ $0 \leq P_{29,2} \leq 15$

$0 \leq P_{0,3} \leq 25$ $\quad$ $0 \leq P_{29,3} \leq 8$

$0 \leq P_{0,4} \leq 20$ $\quad$ $0 \leq P_{24,1} \leq 25$

$0 \leq P_{0,5} \leq 17$

$0 \leq P_{0,6} \leq 17$ $\quad$ $24 \leq P_{25,1} \leq 116$

$0 \leq P_{0,7} \leq 15$ $\quad$ $0 \leq P_{25,2} \leq 56$

$P_i = o \quad \forall i \neq 0,11,24,25,29.$

$-54 \leq Q_o \leq 268$ $\quad$ $-26 \leq Q_{25} \leq 130$

$-15 \leq Q_{11} \leq 74$ $\quad$ $-4 \leq Q_{29} \leq 20$

$-1 \leq Q_{24} \leq 6$

$Q_i = 0 \quad \forall i \neq 0,11,24,25,29.$

$130 \leq U_i \leq 168 \quad \forall i$

$-1 \leq \theta_i \leq 1 \quad \forall i \neq o \; \theta_o = 0$

i	C_i	D_i	i	C_i	D_i	i	C_i	D_i	i	C_i	D_i
0	24,20	18,70	11	2,90	1,93	22	2,70	1,80	33	2,90	1,45
1	-12,00	-9,30	12	12,84	7,50	23	4,42	1,76	34	1,80	0,90
2	0,30	0,30	13	101,00	4,85	24	33,30	11,90	35	0,75	0,75
3	3,28	2,46	14	6,15	2,05	25	96,00	29,60	36	1,70	1,70
4	13,50	9,65	15	3,66	1,83	26	5,14	2,06	37	9,30	4,22
5	6,98	5,41	16	1,25	1,25	27	9,90	1,98	38	2,25	1,50
6	2,02	0,67	17	0,00	0,00	28	-78,70	8,68	39	2,65	1,98
7	-7,40	3,17	18	-67,60	-10,95	29	155,80	10,45	40	28,60	22,70
8	0,00	0,00	19	2,90	0,97	30	4,30	2,15	41	24,80	19,20
9	2,94	1,47	20	5,10	4,08	31	3,14	2,35	42	0,00	0,00
10	1,80	0,90	21	16,10	11,85	32	3,60	4,05	43	0,00	0,00

Sommets reliés i	j	A_{ij}	$1/B_{ij}$	α_{ij}	β_{ij}
10	17	0,9975	0,0330	0,0011	1,1532
9	17	0,9989	0,0489	0,0005	1,1530
17	18	0,9985	0,0426	0,0007	1,1530
18	19	0,9991	0,0528	0,0004	1,1539
19	21	0,9974	0,0315	0,0012	1,1542
20	21	0,9992	0,0550	0,0004	1,1529

20	23	0,9960	0,0254	0,0018	1,1534
15	23	0,9992	0,0550	0,0004	1,1529
15	16	0,9991	0,0508	0,0004	1,1529
16	17	0,9997	0,0826	0,0001	1,1538
23	24	0,9983	0,0375	0,0008	1,1540
24	25	0,9997	0,0997	0,0001	1,3258
25	29	0,9999	0,1904	0,0000	1,3208
13	29	0,9999	0,2090	0,0000	1,3328
13	30	0,9989	0,0523	0,0003	1,3309
29	30	0,9986	0,1026	0,0004	1,3109
12	30	0,9999	0,1829	0,0000	1,3308
11	12	0,9988	0,0504	0,0003	1,3319
26	30	0,9997	0,0975	0,0001	1,3318
26	27	0,9997	0,1126	0,0001	1,3308
27	31	0,9943	0,0255	0,0015	1,3123
30	31	0,9886	0,0179	0,0030	1,3188
14	30	0,9985	0,0457	0,0004	1,3319
14	31	0,9955	0,0261	0,0011	1,3322
22	28	0,9989	0,0523	0,0003	1,3309
21	22	0,9893	0,0231	0,0048	1,1534
8	22	0,9997	0,0912	0,0001	1,1558
7	8	0,9960	0,0252	0,0018	1,1534
6	7	0,9984	0,0400	0,0007	1,1540
1	31	0,9968	0,0692	0,0009	1,3101
0	1	0,9854	0,0147	0,0036	1,3330
5	31	0,9960	0,0249	0,0018	1,1544
4	5	0,9984	0,0388	0,0007	1,1530
3	5	0,9980	0,0367	0,0009	1,1531
2	5	0,9968	0,0281	0,0014	1,1533
0	2	0,9991	0,0532	0,0004	1,1499
0	32	0,9979	0,0345	0,0009	1,1511
32	43	0,9991	0,0528	0,0004	1,1539

33	43	0,9998	0,1207	0,0001	1,1638
42	43	0,9992	0,0601	0,0003	1,1539
41	42	1,0000	0,7246	0,0000	1,5708
40	41	1,0000	1,2046	0,0000	1,5708
35	40	0,9992	0,0550	0,0004	1,1529
34	35	0,9992	0,0574	0,0004	1,1539
39	40	0,9967	0,0275	0,0015	1,1533
38	39	0,9983	0,0377	0,0008	1,1540
37	38	0,9994	0,0629	0,0003	1,1529
36	37	0,9974	0,0314	0,0012	1,1542
0	33	0,9966	0,0670	0,0009	1,3101

Solution donnée.

Optimum : 42,897

Solution

Noeud i	P_i	Q_i	U_i	θ_i
0	332,0003	80,0983	168,0000	
1			151,1937	-0,277889
2			161,7061	-0,067867
3			151,0343	-0,211764
4			149,1469	-0,218833
5			152,3801	-0,208499
6			168,0000	-0,126457
7			167,9445	-0,124352
8			166,7381	-0,129472
9			157,4691	-0,296516
10			157,7165	-0,296964

11	10,000	24,9131	146,6588	-0,510035
12			143,2409	-0,511123
13			140,6144	-0,560693
14			143,8393	-0,453390
15			151,6990	-0,352565
16			155,3127	-0,316410
17			157,6223	-0,294296
18			161,6257	-0,246285
19			160,2794	-0,249201
20			154,4138	-0,292830
21			157,7810	-0,249544
22			166,1719	-0,130287
23			148,6577	-0,384601
24	0,000	-1,0000	141,5250	-0,496141
25	125,1956	41,6381	141,6803	-0,527409
26			142,9333	-0,484921
27			143,3922	-0,462571
28			167,2928	-0,076538
29	23,0000	20,0001	141,0874	-0,550421
30			142,6333	-0,508460
31			147,0452	-0,348081
32			164,5064	-0,022659
33			164,7634	-0,024821
34			155,2233	-0,078516
35			155,2740	-0,077237
36			152,5262	-0,121135
37			152,5952	-0,118790
38			153,1403	-0,112172
39			154,0550	-0,098423
40			155,1855	-0,074877
41			155,3810	-0,073484
42			155,9732	-0,070247
43			162,4973	-0,035827

A N N E X E 2

Influence de la précision des données du Dispatching Economique

A.2.1 GENERALITES.

Les différences constatées entre les solutions trouvées par le code et les solutions obtenues par d'autres méthodes nous ont conduit à supposer qu'une faible variation sur les données du problème pouvait entraîner une variation relativement forte sur la solution. Nous allons vérifier cette hypothèse.

A.2.2 CALCUL D'ERREUR

Considérons la contrainte

$$P_i + \sum_{j\in e(i)} U_i U_j B'_{ij} \cos(\theta_i - \theta_j + \beta_{ij}) - \sum_{j\in e(i)} U_i^2 A_{ij} B'_{ij} \cos(\beta_{ij} - \alpha_{ij}) - C_i \geq 0$$

où $B'_{ij} = \frac{1}{B_{ij}}$.

Nous supposons que les données $A_{ij}, B'_{ij}, \alpha_{ij}, \beta_{ij}$ sont fournies avec une erreur E. Nous allons calculer l'erreur qui en résulte sur la valeur de la contrainte.

Nous obtenons en différentiant

$$D = \sum_{j\in e(i)} [(U_i U_j \cos(\theta_i - \theta_j + \beta_{ij}) - U_i^2 A_{ij} \cos(\beta_{ij} - \alpha_{ij}))\, dB'_{ij}$$

$$- U_i^2 B'_{ij} \cos(\beta_{ij} - \alpha_{ij})\, dA_{ij}$$

$$+ (-U_i U_j B'_{ij} \sin(\theta_i - \theta_j + \beta_{ij}) + U_i^2 A_{ij} B'_{ij} \sin(\beta_{ij} - \alpha_{ij}))\, d\beta_{ij}$$

$$- U_i^2 A_{ij} B'_{ij} \sin(\beta_{ij} - \alpha_{ij})\, d\alpha_{ij}].$$

D'où l'erreur sur la contrainte, en prenant la borne supérieure :

$$E \leqslant \sum_{j \in e(i)} \left[|U_i U_j \cos(\theta_i - \theta_j + \beta_{ij}) - U_i^2 A_{ij} \cos(\beta_{ij} - \alpha_{ij})| \right.$$

$$+ |U_i^2 B'_{ij} \cos(\beta_{ij} - \alpha_{ij})|$$

$$+ |-U_i U_j B'_{ij} \sin(\theta_i - \theta_j + \beta_{ij}) + U_i^2 A_{ij} B'_{ij} \sin(\beta_{ij} - \alpha_{ij})|$$

$$\left. + |U_i^2 A_{ij} B'_{ij} \sin(\beta_{ij} - \alpha_{ij})| \right] \times E.$$

A.2.3 APPLICATION NUMERIQUE.

Considérons le problème N°5 (voir annexe A1). Le calcul effectif sur ordinateur de la valeur ci-dessus, pour la 1ère contrainte de ce problème, donne :

$$E \leqslant 0{,}2.10^5 \times E.$$

Dans ce problème, les données sont fournies avec 4 décimales. Donc

$$E \leqslant 5.10^{-5}.$$

D'où finalement

$$E \leqslant 1.$$

On voit donc que suivant la précision des données, l'optimum du problème peut varier considérablement.

A.2.4 CALCUL DES DONNEES.

Si l'on veut obtenir un optimum précis, il est nécessaire d'avoir une bonne précision sur les données du problème : $A_{ij}, B'_{ij}, \alpha_{ij}, \beta_{ij}$.

Ces valeurs sont calculées à partir des R, X, g, h, données "brutes" du problème, de la manière suivante :

Soient $\bar{Z} = R + jX$

$\bar{Y} = g + jh.$

D'où $\bar{A} = \text{ch } \sqrt{\bar{Y}\bar{Z}}$

$$\bar{B} = \sqrt{\frac{\bar{Z}}{\bar{Y}}} \text{ sh } \sqrt{\bar{Y}\ \bar{Z}}$$

D'où les valeurs de A,B,α,β

$A = |\text{ch } \sqrt{\bar{Y}\ \bar{Z}}|$

$B = |\sqrt{\frac{\bar{Z}}{\bar{Y}}} \text{ sh } \sqrt{\bar{Y}\ \bar{Z}}|$

$\alpha = \text{Arg } \bar{A}$

$\beta = \text{Arg } \bar{B}$.

Ce calcul doit être effectué pour chacune des lignes du réseau.

Application numérique :

Ces calculs ont été effectués, pour le problème N°5, sur IBM 1620 en double précision (18 chiffres significatifs).

Données brutes :

Noeuds reliés i	j	R	X	h
1	2	5	24,5	0,0004
1	3	5,75	28	0,0004
1	4	6	39,5	0,0006
1	8	22,8	62,6	0,0004
2	4	5	24,5	0,0002
3	4	24,7	97	0,0004
3	5	24,7	97	0,0004
4	6	3,75	24,75	0,0002
4	7	8,25	33	0,0006
4	10	2	10	0,0004
5	6	9,5	31,8	0,0004
5	9	6	39,5	0,0006
7	8	8,25	32,3	0,0006

Résultats obtenus sur IBM 1620 :

g = 0 pour toutes les lignes.

Sommets reliés i	j	A_{ij}	$1/B_{ij}$	α_{ij}	β_{ij}
1	2	0,995104334677	0,040057396538	0,001003279343	1,369812769732
1	3	0,994405667037	0,035049621618	0,001154312381	1,368638634621
1	4	0,988174473025	0,025128486632	0,001814355044	1,420650895803
1	8	0,987513087074	0,015072689920	0,004598428541	1,223034914400
2	4	0,997551083709	0,040024680992	0,000500818248	1,369645939556
3	4	0,980670876368	0,010055361828	0,005004875597	1,323107111138
3	5	0,980670876368	0,010055361828	0,005004875597	1,323107111138
4	6	0,997526067714	0,039981077282	0,000375619968	1,420549940057
4	7	0,990118378327	0,029495492372	0,002491462657	1,326643754683
4	10	0,998000719975	0,098123469785	0,000400534177	1,373534135847
5	6	0,993647946692	0,030194704289	0,001908095963	1,281127803991
5	9	0,988174473025	0,025128486632	0,001814355044	1,420650895803
7	8	0,990327693087	0,030093846661	0,002491110663	1,321551014520

A N N E X E 3

Expériences numériques

A.3.1. PROBLEME N°1

A.3.1.1 Valeurs des variables θ_1 et θ_2 en fonction du numéro de troncature, pour $k = 10^{-2}$

Troncature	θ_1	θ_2
10	0,1182	-0,3969
11	0,1211	-0,3955
12	0,1183	-0,3968
13	0,1208	-0,3956
14	0,1184	-0,3967
15	0,1205	-0,3957
16	0,1185	-0,3967
17	0,1203	-0,3958
18	0,1185	-0,3966
19	0,1202	-0,3958
20	0,1185	-0,3967

A.3.1.2 Valeurs de la fonction économique en fonction du numéro de troncature, avec centrage et $k = 10^{-1}$.

0	7596,75
1	5381,22
2	5149,32
3	5128,25
4	5126,39
5	5126,217
6	5126,202
7	5126,200389
8	5126,200262
9	5126,2002501
10	5126,20024889
11	5126,20024854

A.3.2 PROBLEME N°2.

Valeurs de la fonction économique en fonction du numéro de troncature, sans centrage et avec $k = 10^{-2}$

0	8880
1	5226,80
2	5174,861
3	5174,1363
4	5174,12620
5	5174,126063
6	5174,126061

A.3.3 PROBLEME N°3.

A.3.3.1 Valeur de la fonction économique en fonction du numéro de troncature, avec $k = 10^{-2}$.

70	5362,675
71	5362,6627
72	5362,6507
73	5362,6428
74	5362,63094
75	5362,61941
76	5362,60845
77	5362,597646
78	5362,587001
79	5362,576589

A.3.3.2 Valeurs des variables U_2, θ_1 et θ_2 en fonction du numéro de troncature.

(les variables U_0 et U_1 sont constamment à leur borne supérieure 252).

Troncature	U_2	Θ_1	Θ_2
60	201,425	0,1333	-0,3714
61	201,415	0,1340	-0,3711
62	201,426	0,1334	-0,3714
63	201,417	0,1340	-0,3711
64	201,427	0,1334	-0,3714
65	201,418	0,1340	-0,3711

A.3.4 PROBLEME N°4

A.3.4.1 Valeurs des variables x_3 et x_6 en fonction du numéro de troncature, pour $k = 10^{-2}$

Troncature	x_3	x_6
10	372,55	0,1555
11	373,34	0,1507
12	372,85	0,1549
13	373,44	0,1513
14	373,04	0,1546
15	373,50	0,1516
16	373,16	0,1544
17	373,54	0,1519
18	373,26	0,1542

A.3.4.2 Valeurs de la fonction économique en fonction du numéro de troncature, avec centrage et $k = 10^{-4}$.

0	41489,9999	7	8827,7301
1	29338,2151	8	8827,6272
2	11166,4348	9	8827,60465
3	9072,0033	10	8827,599401
4	8856,7836	11	8827,5981373
5	8831,5850	12	8827,5981250
6	8828,2631	13	8827,59812356

A.3.5 PROBLEME N°5

A.3.5.1 - Valeurs des variables P_8, P_9, P_{10} en fonction du numéro de troncature, avec lin = 1, dic = 10^4, k = 10^{-5}, et 2 contraintes de centrage.

Troncature	P_8	P_9	P_{10}
1	480	200	460
2	304,93	200	224,24
3	378,62	145,28	323,47
4	266,14	173,44	239,37
5	317,55	148,57	292,40
6	391,49	171,98	368,69
7	360,56	158,36	340,81
8	320,60	140,75	304,78
9	295,52	150	329,02
10	320	139,39	306,59
11	333,99	144,69	320
12	319,99	149,13	331,26
13	333,25	148,14	311,59
14	319,96	152,08	322,90
15	322,81	150,57	319,89

Les valeurs optimales de ces 3 variables sont respectivement 320,150, 320.

A.3.5.2 - Valeurs de la fonction économique en fonction du numéro de troncature.

0	4569,90	7	1518,13
1	2949,54	8	1506,72
2	1808,21	9	1501,07
3	1636,91	10	1499,82
4	1588,88	11	1499,36
5	1558,65	12	1499,26
6	1536,83	13	1499,24

A.3.5.3 Valeurs de la fonction économique en fonction du numéro de troncature.

1	1499,964
2	1499,652
3	1499,3918
4	1499,2642
5	1499,24137
6	1499,23666
7	1499,236151
8	1499,235946
9	1499,2358817
10	1499,23583102
11	1499,23582977

A.3.6 PROBLEME N°6

Valeurs de la fonction économique en fonction du numéro de troncature.

1	84,442
2	68,242
3	57,848
4	50,628
5	46,5138
6	44,4991
7	43,55100
8	43,15719
9	42,996397
10	42,935392
11	42,9121362
12	42,9033652
13	42,90029257
14	42,89901968
15	42,898487771
16	42,898291565
17	42,8982166678
18	42,8981887736
19	42,8981784084

A N N E X E 4

Code ALGOL 60 de la Méthode des centres linéarisée adaptée au problème du Dispatching Economique

A.4.1 PRINCIPALES VARIABLES ENTIERES.

ADC nombre de contraintes additionnelles dans le centrage

AFF précision demandée sur la recherche du point-frontière dans le centrage (nombre de réductions de l'intervalle)

BSAC nombre d'éléments non nuls dans la matrice des programmes linéaires

CTR numéro de la contrainte à linéariser (si CTR = 0, toutes les contraintes sont linéarisées).

F1 nombre de variables P_i

ICEN numéro de troncature

ILIN numéro de linéarisation

ILINMA nombre de linéarisations par troncature

ITMAX nombre maximum de troncatures

JCEN numéro de troncature à partir duquel commence le centrage

M1 nombre de contraintes

N nombre de variables

NE nombre de noeuds du réseau

NLN nombre de lignes du réseau

NNUL dimension des tableaux utilisés dans la résolution des systèmes par élimination de Gauss.

A.4.2 PRINCIPALES VARIABLES REELLES.

DELTA coefficient utilisé pour définir la fonction économique dans la méthode simpliciale avec calculs directs (phase mixte)

DIC précision demandée pour la recherche du maximum de la F-distance sur un segment

EP1, EP2, EP3,EP4

précision demandée sur les calculs de la méthode simpliciale

FCT 1 valeur de la fonction économique à chaque troncature

FCT 2 valeur de la fonction économique à chaque linéarisation

KKK valeur du coefficient de pondération de la norme de la fonction économique

PREC précision demandée sur l'optimum du problème

A.4.3 TABLEAUX ENTIERS.

BB contient la base dans la méthode simpliciale

RP sert à réordonner implicitement les lignes de la matrice des programmes linéaires

ILI contient les indices de lignes des éléments de AC

ICO contient les indices de colonnes des éléments de AC

CH contient les chaînons associés aux éléments de AC

TC contient les têtes de chaînes (une par colonne)

ITAG, LNXT
utilisés dans la résolution des systèmes par élimination de Gauss ([38]).

RG1, RG2
contiennent les indices des noeuds extrémités de chacune des lignes du réseau

P1 donne le nombre d'usines productrices en chaque noeud du réseau

A.4.4 TABLEAUX REELS.

VCONT donne la valeur de la fonction économique et des contraintes

NOR contient la norme de la fonction économique et des contraintes

AC matrice condensée contenant les éléments non nuls de la matrice des programmes linéaires

A second membre des programmes linéaires

XO bornes inférieures sur les variables

X1 bornes supérieures sur les variables

Z optimum des programmes linéaires

YD point de linéarisation

X4 point-frontière dans le centrage

CE, RE utilisés dans la résolution des systèmes par élimination de Gauss([38]).

C1 consommations de puissance active en chaque noeud

D1 consommations de puissance réactive en chaque noeud

GD valeurs des coefficients de la fonction économique

A.4.5 TABLEAUX BOOLEENS.

BM contient "vrai" <==> la variable correspondante appartient à B^-

BP contient "vrai" <==> la variable correspondante appartient à B^+

VNRM contient "vrai" <==> la valeur de la variable correspondante est inférieure à sa borne inférieure

VNRP contient "vrai" <==> la valeur de la variable correspondante est supérieure à sa borne supérieure

Q2 contient "vrai" <==> le noeud correspondant est producteur

A.4.6 PROCEDURES.

FECON (V,X)
calcul de la fonction économique au point X (résultat dans V [1]).

CALCT (II,U,X)
calcul au point X de la valeur de la contrainte II (résultat dans U).

CONT (PP,FDIST,X)
calcul au point X de toutes les contraintes (résultat dans VCONT)
- si PP = 1, calcul de la F-distance (résultat dans FDIST)

- si PP = 2, linéarisation de la contrainte CTR (de toutes les contraintes si CTR = 0)

ELA (I,J) donne la valeur de l'élément situé à l'intersection de la ligne I et de la colonne J de la matrice des programmes linéaires

RZMAX (YH,ZH,XM)
recherche du zéro (résultat dans X4) et du maximum (résultat dans XM) de la F-distance sur le segment [YH,ZH] par interpolation polygonale

REDUCT (N,DE,LCOL,NOZE,NSEQ,BB,RP,ERR)
factorisation E.F.I. d'une matrice symétrique en structure ([38]).

SOLVE (V,N,N1,DE,LCOL,NOZE,NSEQ,BB,RP,TRANSP)
résolution d'un système avec une matrice de la forme B de V.4.1, avec V pour second membre, si TRANSP = faux ; avec la transposée de la matrice si TRANSP = vrai

SYST (X,NS,NB1,NB2,DE,LCOL,NOZE,NSEQ,BB,RP,TRANSP)
résolution d'un système avec la matrice de base A^I et X pour second membre, si TRANSP = faux ; avec la transposée de A^I si TRANSP = vrai.

SIMPB (BM,BP,BB,RP,M,NP,NB2,DELTA,ERR)
méthode simpliciale en variables bornées, avec calculs directs en partant d'une base quelconque BB.

```
'BEGIN''INTEGER'ADC,AFF,BIDULE,BSAC,CTR,F,F1,I,I1,I2,ICEN,II,ILIN,
ILINMA,IP,ITMAX,J,JCEN,JJ,K,KP,L,LA,LF,LF1,LI,LK,LM,LP,LR,LR1,M1,MIN,
MM2,N,NB1,NB2,NE,NLN,NN,NNUL,NP,NP1,NQ,NS,NS1,NS2,NB12,R,S,SM,SP;
'REAL'AA,CF,CIJ,D,DELTA,DIC,AIJ,EP1,EP2,EP3,EP4,FCT1,FCT2,
FCT3,FCT5,FDIST,KKK,MAX,PREC,RF,SIJ,SUM,T,T1,TI,TJ,U,U1,
UM,UP,UPUI,XI,XJ;
'BOOLEAN'BOOL,BOOL1,TEST;
BIDULE:=1;UPUI:='E'50;FCT1:=UPUI;
FRM1:'FORMAT'(11I4);FRM2:'FORMAT'(8E10.0);
'READ'(105,FRM1)AFF,ICEN,ADC,N,M1,ITMAX,ILINMA,LP,F1,NE,NLN;
'READ'(105,FRM2)U,KKK,PREC,DIC,EP1,EP2,EP3,EP4;
M1:=M1+1;EP2:=-EP2;EP3:=-EP3;NN:=N+1;NP:=NN+M1;NQ:=NN+1;
F1RM:'FORMAT'(2I4,F5.1);'READ'(105,F1RM)NNUL,BSAC,DELTA;
'BEGIN''ARRAY'A(/1:M1+ADC/),CE,RE(/1:NNUL/),X1,XO,Z(/1:NN/),XC,X2,
X4,X5,YD(/1:N/),VCONT(/1:M1/),NOR(/1:M1+ADC/),C1,D1(/1:NE/),B1,B2,CS1,
SN1(/1:NLN/),GD(/1:F1/),AC(/1:BSAC/);
'INTEGER''ARRAY'ITAG,LNXT(/1:NNUL/),BB,RP,BB2,RP2(/1:M1+ADC/),RG1,
RG2(/1:NLN/),P1(/1:NE/),ILI,ICO,CH(/1:BSAC/),TC(/1:N/);
'BOOLEAN''ARRAY'BM1,BM2,BP,BP2,VNRM,VNRP(/1:NP+ADC/),Q2(/1:NE/);

'PROCEDURE'FECON(V,X);'ARRAY'V,X;'BEGIN''REAL'U;'INTEGER'I;U:=0;
'FOR'I:=1'STEP'1'UNTIL'F1'DO'U:=U+GD(/I/)*X(/2*NE-1+I/));
V(/1/):=2371.6-U'END';

'PROCEDURE' CALCT(II,U,X); 'VALUE' II; 'INTEGER' II; 'REAL' U;
'ARRAY'X;'BEGIN' 'INTEGER' I,J,K; I1:=II;
'IF' II=1 'THEN' 'BEGIN' FECON(VCONT,X);
U:=VCONT(/1/)-FCT1; U:=KKK*NOR(/1/)*U; 'GOTO' BN9 'END';
I:=II-1; TEST:='TRUE'; 'IF' I>NE 'THEN' 'BEGIN' I:=I-NE;
TEST:='FALSE' 'END'; 'IF' TEST 'THEN' 'BEGIN' J:=2*NE-1; U:=-C1(/I/);
'FOR' K:=1 'STEP' 1 'UNTIL' I-1 'DO' J:=J+P1(/K/);
'FOR' K:=1 'STEP' 1 'UNTIL' P1(/I/) 'DO' U:=U+X(/J+K/) 'END' 'ELSE'
'BEGIN'U:=-D1(/I/); 'IF' Q2(/I/) 'THEN' 'BEGIN' J:=2*NE-1+F1;
'FOR' K:=1 'STEP' 1 'UNTIL' I 'DO' 'IF' Q2(/K/) 'THEN' J:=J+1;
U:=U+X(/J/)'END''END';'FOR'K:=1'STEP'1'UNTIL'NLN'DO''BEGIN''IF'I=RG1(
/K/)'THEN' J:=RG2(/K/) 'ELSE' 'IF' I=RG2(/K/) 'THEN' J:=RG1(/K/)
'ELSE''GOTO'BN8;XI:=X(/I/); XJ:=X(/J/); 'IF' I=NE 'THEN' TI:=0 'ELSE'
TI:=X(/I+NE/);'IF'J=NE'THEN'TJ:=0'ELSE'TJ:=X(/J+NE/);AIJ:=TI-TJ+B2(/K/
);'IF' TEST 'THEN' U:=U-XI*(XI*CS1(/K/)-XJ*B1(/K/)*COS(AIJ))
'ELSE' U:=U-XI*(XI*SN1(/K/)-XJ*B1(/K/)*SIN(AIJ));
BN8: 'END'; U:=NOR(/I/)*U; BN9: 'END' CALCT;

'PROCEDURE'CONT(PP,FDIST,X);'VALUE'PP;'INTEGER'PP;'REAL'FDIST;
'ARRAY'X;'BEGIN''INTEGER'I,J,K,KC;KC:=2*NE;
'FOR'I:=1'STEP'1'UNTIL'NE'DO''BEGIN'VCONT(/I+1/):=-C1(/I/);
'FOR'K:=1'STEP'1'UNTIL'P1(/I/)'DO''BEGIN'VCONT(/I+1/):=VCONT(/I+1/)+
X(/KC/);KC:=KC+1'END''END';
'FOR'I:=1'STEP'1'UNTIL'NE'DO''BEGIN'VCONT(/I+NE+1/):=-D1(/I/);
'IF'Q2(/I/)'THEN''BEGIN'VCONT(/I+NE+1/):=VCONT(/I+NE+1/)+X(/KC/);
KC:=KC+1'END''END';
'FOR'K:=1'STEP'1'UNTIL'NLN'DO''BEGIN'TEST:='TRUE';
BN:'IF'TEST'THEN''BEGIN'I:=RG1(/K/);J:=RG2(/K/)'END''ELSE''BEGIN'
KC:=I;I:=J;J:=KC'END';XI:=X(/I/);XJ:=X(/J/);
'IF'I=NE'THEN'TI:=0'ELSE'TI:=X(/I+NE/);
'IF'J=NE'THEN'TJ:=0'ELSE'TJ:=X(/J+NE/);
```

```
AIJ:=TI-TJ+B2(/K/);SIJ:=B1(/K/)*SIN(AIJ);
CIJ:=B1(/K/)*COS(AIJ);
VCONT(/I+1/):=VCONT(/I+1/)-XI*(XI*CS1(/K/)-XJ*CIJ);
VCONT(/I+NE+1/):=VCONT(/I+NE+1/)-XI*(XI*SN1(/K/)-XJ*SIJ);
'IF'TEST'THEN''BEGIN'TEST:='FALSE';'GOTO'BN'END''END';
FECON(VCONT,X);'IF'PP=3'THEN''GOTO'E12;
'IF'PP=1'THEN''BEGIN'VCONT(/1/):=VCONT(/1/)-FCT1;
'FOR'J:=1'STEP'1'UNTIL'M1'DO'VCONT(/J/):=VCONT(/J/)*NOR(/J/);
VCONT(/1/):=KKK*VCONT(/1/);U:=VCONT(/1/);
U1:=VCONT(/2/);'IF'U>U1'THEN''BEGIN'T:=U;U:=U1;U1:=T;I1:=2;I2:=1'END'
'ELSE''BEGIN'I1:=1;I2:=2'END';'FOR'J:=3'STEP'1'UNTIL'M1'DO''BEGIN''IF'
U1>VCONT(/J/)'THEN''BEGIN'U1:=VCONT(/J/);I2:=J;'IF'U>U1'THEN''BEGIN'
T:=U;U:=U1;U1:=T;I2:=I1;I1:=J'END''END''END';
FDIST:=U;'GOTO'E12'END';'IF'CTR=0'OR'CTR=1'THEN''BEGIN'AA:=0;
'FOR'K:=1'STEP'1'UNTIL'F1'DO''BEGIN'AC(/LR1/):=U:=GD(/K/);ILI(/LR1/):=F;
ICO(/LR1/):=K+2*NE-1;TC(/K+2*NE-1/):=LR1;LR1:=LR1+1;
AA:=AA+U*U'END';NOR(/F/):=1/SQRT(AA);F:=F+1'END';KC:=2;
TEST:='TRUE';BN2:'FOR'I:=1'STEP'1'UNTIL'NE'DO''BEGIN''IF'CTR=0
'OR'CTR=KC'THEN''BEGIN'AA:=U:=T:=0;'IF'TEST'THEN''BEGIN'J:=2*NE-1;
'FOR'K:=1'STEP'1'UNTIL'I-1'DO'J:=J+P1(/K/);'FOR'K:=1'STEP'1'UNTIL'
P1(/I/)'DO''BEGIN'AC(/LR1/):=-1;ILI(/LR1/):=F;ICO(/LR1/):=J+K;
CH(/LR1/):=TC(/J+K/);TC(/J+K/):=LR1;LR1:=LR1+1;AA:=AA+1'END''END''ELSE'
'BEGIN''IF'Q2(/I/)'THEN''BEGIN'J:=2*NE-1+F1;'FOR'K:=1'STEP'1'UNTIL'I'DO'
'IF'Q2(/K/)'THEN'J:=J+1;AC(/LR1/):=-1;ILI(/LR1/):=F;ICO(/LR1/):=J;CH(/LR
1/):=TC(/J/);TC(/J/):=LR1;LR1:=LR1+1;AA:=AA+1'END''END';'FOR'K:=1'STEP'1
'UNTIL'NLN
'DO''BEGIN''IF'I=RG1(/K/)'THEN'J:=RG2(/K/)'ELSE''IF'I=RG2(/K/)'THEN'J:=
RG1(/K/)'ELSE''GOTO'BN1;XI:=X(/I/);XJ:=X(/J/);'IF'I=NE'THEN'TI:=0'ELSE'
TI:=X(/I+NE/);'IF'J=NE'THEN'TJ:=0'ELSE'TJ:=X(/J+NE/);
AIJ:=TI-TJ+B2(/K/);SIJ:=-B1(/K/)*COS(AIJ);
CIJ:=B1(/K/)*SIN(AIJ);'IF'TEST'THEN''BEGIN'AIJ:=-SIJ;
SIJ:=CIJ;CIJ:=AIJ;AIJ:=CS1(/K/)'END''ELSE'AIJ:=SN1(/K/);
U:=U+2*XI*AIJ-XJ*CIJ;U1:=XI*CIJ;AC(/LR1/):=-U1;ILI(/LR1/):=F;ICO(/LR1/)
:=J;CH(/LR1/):=TC(/J/);TC(/J/):=LR1;LR1:=LR1+1;AA:=AA+U1*U1;
'IF'I<NE'THEN'T:=T+XI*XJ*SIJ;'IF'J<NE'THEN''BEGIN'T1:=XI*XJ*SIJ;
AC(/LR1/):=-T1;ILI(/LR1/):=F;ICO(/LR1/):=J+NE;CH(/LR1/):=TC(/J+NE/);
TC(/J+NE/):=LR1;LR1:=LR1+1;AA:=AA+T1*T1'END';BN1:'END';
AC(/LR1/):=U;ILI(/LR1/):=F;ICO(/LR1/):=I;CH(/LR1/):=TC(/I/);TC(/I/):=
LR1;LR1:=LR1+1;
AA:=AA+U*U;'IF'I<NE'THEN''BEGIN'AC(/LR1/):=T;ILI(/LR1/):=F;ICO(/LR1/)
:=I+NE;CH(/LR1/):=TC(/I+NE/);TC(/I+NE/):=LR1;LR1:=LR1+1;AA:=AA+T*T'END';
AA:=1/SQRT(AA);NOR(/F/):=AA;F:=F+1'END';KC:=KC+1'END';
'IF'TEST'THEN''BEGIN'TEST:='FALSE';'GOTO'BN2'END';E12:'END'CONT;

'REAL''PROCEDURE'ELA(I,J);'VALUE'I,J;'INTEGER'I,J;'BEGIN''REAL'U;
'INTEGER'K;U:=0;'IF'I<NN'THEN''BEGIN''FOR'K:=TC(/J/),CH(/K/)'WHILE'
K'NE'0'DO''IF'I=ILI(/K/)'THEN'U:=AC(/K/)'END''ELSE''IF'J=NN'THEN'U:=1
'ELSE''IF'I=J-NN'THEN'U:=1;ELA:=U'END';

'PROCEDURE' RZMAX(YH,ZH,XM);'ARRAY' YH,ZH,XM;
'BEGIN' 'REAL' VMIN,VMIN1,A1,B,C,D,E,FA,FB,FC,FD,FE,FA2,FB2,
FC2,FD2,FE2;
'INTEGER' IB1,IA1,IA2,IB2,IC1,IC2,ID1,ID2,IE1,IE2,PP;
'ARRAY' XC,T(/1:N/);VMIN:=0;R:=0;PP:=1;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO' 'BEGIN' B:=ZH(/I/)-YH(/I/);
```

```
T(/I/):=B;VMIN:=VMIN+B*B 'END'; VMIN:=SQRT(VMIN); A1:=0;
B:=D:=VMIN; IB1:=1; C:=1/VMIN;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO' T(/I/):=T(/I/)*C;
VMIN:=VMIN/DIC;VMIN1:=VMIN/10;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO' X4(/I/):=YH(/I/);
C:=0; 'IF' BIDULE=0 'THEN'
'BEGIN' CONT(PP,FA,YH); BOOL1:='TRUE'; IA1:=I1; 'IF' FA>=EP1
'THEN' 'GOTO' RMAX; RZ1: C:=(A1+B)*.5;
RZ2: 'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO' X4(/I/):=YH(/I/)+C*T(/I/);
'IF' C-A1<VMIN1 'OR' B-C<VMIN1 'THEN' 'GOTO' RMAX;
'IF' BOOL1 'THEN' CONT(PP,FC,X4) 'ELSE' CALCT(IA1,FC,X4);
'IF' FC>0 'THEN' 'BEGIN' B:=C; FB:=FC; IB1:=I1 'END''ELSE''BEGIN'
A1:=C; FA:=FC; IA1:=I1 'END'; 'IF' IA1=IB1 'THEN' 'BEGIN'
C:=(B*FA-A1*FB)/(FA-FB); BOOL1:='FALSE'; 'GOTO' RZ2 'END'
'ELSE''BEGIN'BOOL1:='TRUE';'GOTO'RZ1'END''END';RMAX:A1:=C;
B:=D; CONT(PP,FA,X4); IA1:=I1; IA2:=I2; FA2:=VCONT(/IA2/);
CONT(PP,FB,ZH); IB2:=I2; IB1:=I1; FB2:=VCONT(/IB2/);
RET: 'IF' IA1=IB2 'AND' IA2=IB1 'THEN' 'GOTO' RET2;
C:=(A1+B)*.5; 'IF' B-A1<VMIN 'THEN' 'GOTO' FINI;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO' XC(/I/):=YH(/I/)+C*T(/I/);
CONT(PP,FC,XC);IC1:=I1; IC2:=I2; FC2:=VCONT(/IC2/); 'IF' FC<FB 'THEN'
'BEGIN'A1:=C; FA:=FC; IA1:=IC1; IA2:=IC2; FA2:=FC2; 'GOTO' RET 'END'
'ELSE' 'IF' FC<FA 'THEN' 'BEGIN' B:=C; FB:=FC; IB1:=IC1; IB2:=IC2;
FB2:=FC2; 'GOTO' RET 'END'; RET1: 'IF' IA1=IB2 'AND' IA2=IB1
'THEN' 'GOTO' RET2; RET3: 'IF' B-A1 < VMIN 'THEN' 'GOTO' FINI;
D:=(A1+C)*.5; 'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
XC(/I/):=YH(/I/)+D*T(/I/);
CONT(PP,FD,XC); ID1:=I1; ID2:=I2; FD2:=VCONT(/ID2/);
'IF' FD>=FA 'THEN' 'BEGIN' 'IF' FD>=FC 'THEN' 'BEGIN'
B:=C;FB:=FC;IB1:=IC1;IB2:=IC2; FB2:=FC2; C:=D; FC:=FD; IC1:=ID1;
IC2:=ID2;FC2:=FD2 'END' 'ELSE' 'BEGIN' E:=(B+C)*.5;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
XC(/I/):=YH(/I/)+E*T(/I/); CONT(PP,FE,XC); IE1:=I1; IE2:=I2;
FE2:=VCONT(/IE2/); 'IF' FE>FC 'THEN' 'BEGIN' A1:=C;
FA:=FC;IA1:=IC1;IA2:=IC2; FA2:=FC2; C:=E; FC:=FE; IC1:=IE1; IC2:=IE2;
FC2:=FE2 'END' 'ELSE' 'IF' FE<FB 'THEN' 'GOTO' FINI 'ELSE' 'BEGIN'
A1:=D;FA:=FD; IA1:=ID1; IA2:=ID2; FA2:=FD2; B:=E; FB:=FE;
IB1:=IE1; IB2:=IE2; FB2:=FE2 'END' 'END', 'GOTO' RET1
'END''ELSE''GOTO'FINI;RET2: D:=A1; E:=B; FD:=FA; FD2:=FA2; FE:=FB;
FE2:=FB2;RET7: U:=FA-FA2; U1:=FB2-FB; C:=(B*U-A1*U1)/(U-U1);
'IF' C-A1<VMIN 'OR' B-C < VMIN 'THEN' 'GOTO'
RET5; 'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
XC(/I/):=YH(/I/)+C*T(/I/); CALCT(IA1,FC,XC);
CALCT(IB1,FC2,XC); RET4: 'IF' FC<FC2 'THEN' 'BEGIN'
A1:=C;FA:=FC;FA2:=FC2 'END' 'ELSE' 'BEGIN' B:=C; FB:=FC2; FB2:=FC
'END';'GOTO'RET7;RET5: 'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
XC(/I/):=YH(/I/)+C*T(/I/);
CONT(PP,FC,XC); FC2:=VCONT(/I2/); 'IF' 'NOT'((IA1=I1 'AND'
IA2=I2)'OR' (IB1=I1 'AND' IB2=I2)) 'THEN' 'BEGIN' A1:=D;
B:=E;FA:=FD;FA2:=FD2; FB:=FE; FB2:=FE2; IC1:=I1; IC2:=I2;
'GOTO'RET3'END';'IF' ICEN>1 'AND' FC<EP2 'THEN' 'BEGIN' R:=R+1;
VMIN:=VMIN*.1;'IF' R<=6 'THEN' 'GOTO' RET4 'END';
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
XM(/I/):=XC(/I/); 'GOTO' RET6;
FINI: 'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
```

```
XM(/I/):=YM(/I/)+C*T(/I/); CONT(PP,FC,XM);RET6:'IF'FCT5<FC
'THEN' 'BEGIN' FCT5:=FC; 'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
X5(/I/):=XM(/I/)'END'; BOOL1:=FC>=0;
FRM3: 'FORMAT'(1H ,7HFDIST  ,E18.12);
'WRITE'(105,FRM3)FC; 'END' RZMAX;

'PROCEDURE'REDUCT(N,DE,LCOL,NOZE,NSEQ,BB,RP,ERR);
'VALUE'N;'INTEGER'N;'ARRAY'DE;'INTEGER''ARRAY'LCOL,NOZE,NSEQ,BB,RP;
'LABEL'ERR;'BEGIN''INTEGER'M;'FOR'I:=1'STEP'1'UNTIL'NNUL-1'DO'
LNXT(/I/):=I+1;LF:=1;'FOR'I:=1'STEP'1'UNTIL'N'DO''BEGIN'K:=1;
LF1:=LF;LCOL(/I/):=LF;NSEQ(/I/):=I;'FOR'J:=1'STEP'1'UNTIL'N'DO'
'BEGIN'II:=RP(/I/);JJ:=BB(/J/);'IF'I=J'THEN'DE(/I/):=ELA(II,JJ)
'ELSE''IF'ELA(II,JJ)'NE'0'THEN''BEGIN'RE(/LF/):=ELA(II,JJ);
CE(/LF/):=ELA(RP(/J/),BB(/I/));ITAG(/LF/):=J;K:=K+1;LF:=LF+1'END''END';
'IF'LF=LF1'THEN'LCOL(/I/):=0'ELSE'LNXT(/LF-1/):=0;NOZE(/I/):=K'END';
'FOR'I:=LF'STEP'1'UNTIL'NNUL'DO''BEGIN'ITAG(/I/):=0;CE(/I/):=0;RE(/I/):=
0'END';LNXT(/NNUL/):=0;'FOR'J:=1'STEP'1'UNTIL'N-1'DO''BEGIN'
K:=NSEQ(/J/);MIN:=NOZE(/K/);M:=J;'FOR'I:=J+1'STEP'1'UNTIL'N'DO'
'BEGIN'K:=NSEQ(/I/);'IF'NOZE(/K/)<MIN'THEN''BEGIN'MIN:=NOZE(/K/);
M:=I'END''END';KP:=NSEQ(/M/);NSEQ(/M/):=NSEQ(/J/);NSEQ(/J/):=KP;
LK:=LCOL(/KP/);RET6:'IF'LK<=0'THEN''GOTO'FINJ;K:=ITAG(/LK/);LA:=0;
LI:=LCOL(/KP/);IP:=ITAG(/LI/);L:=LCOL(/K/);I:=ITAG(/L/);
RET5:'IF'I>IP'THEN''GOTO'SUP;'IF'I=IP'THEN''BEGIN'LA:=L;L:=LNXT(/L/);
'IF'L<=0'THEN'I:=N+1'ELSE'I:=ITAG(/L/);'GOTO'BRANCH'END''ELSE''BEGIN'
'IF'I=KP'THEN''BEGIN'LM:=LNXT(/L/);'IF'LA<=0'THEN'LCOL(/K/):=LM'ELSE'
LNXT(/LA/):=LM;LNXT(/L/):=LF;LF:=L;CE(/L/):=0;RE(/L/):=0;
NOZE(/K/):=NOZE(/K/)-1;L:=LM'END''ELSE''BEGIN'LA:=L;LI:=LNXT(/L/)
'END';'IF'I>0'THEN''BEGIN'I:=ITAG(/L/);'GOTO'RET5'END''ELSE''IF'LI>0
'THEN''BEGIN'I:=N+1;'GOTO'RET5'END''ELSE''GOTO'BAS'END';SUP:'IF'IP'NE'K
'THEN''BEGIN''IF'LF<=0'THEN''BEGIN''WRITE'(105,FRMM3);FRMM3:'FORMAT'
(27H DIMENSIONED AREA TOO SMALL);'GOTO'ERR'END''ELSE''BEGIN'
LM:=LF;'IF'LA<=0'THEN'LCOL(/K/):=LM'ELSE'LNXT(/LA/):=LM;
LF:=LNXT(/LM/);LNXT(/LM/):=L;ITAG(/LM/):=IP;NOZE(/K/):=NOZE(/K/)+1;
LA:=LM'END''END';BRANCH:LI:=LNXT(/LI/);'IF'LI>0'THEN''BEGIN'
IP:=ITAG(/LI/);'GOTO'RET5'END''ELSE''IF'I>0'THEN''BEGIN'IP:=N+1;
'GOTO'RET5'END';BAS:LK:=LNXT(/LK/);'GOTO'RET6;FINJ:'END';
'FOR'J:=1'STEP'1'UNTIL'N'DO''BEGIN'KP:=NSEQ(/J/);D:=1/DE(/KP/);
DE(/KP/):=D;LK:=LCOL(/KP/);'IF'LK<=0'THEN''GOTO'FNJ;
RES:RE(/LK/):=D*RE(/LK/);LK:=LNXT(/LK/);'IF'LK>0'THEN''GOTO'RES;
LK:=LCOL(/KP/);REY:K:=ITAG(/LK/);CF:=RE(/LK/);RF:=CE(/LK/);
LI:=LCOL(/KP/);IP:=ITAG(/LI/);L:=LCOL(/K/);REV:'IF'L>0'THEN'
I:=ITAG(/L/)'ELSE'I:=N+1;REX:'IF'I>IP'THEN''GOTO'SUS;
'IF'I<IP'THEN''BEGIN'L:=LNXT(/L/);'GOTO'REV'END''ELSE''BEGIN'
CE(/L/):=CE(/L/)-CF*CE(/LI/);RE(/L/):=RE(/L/)-RF*RE(/LI/);L:=LNXT(/L/);
'IF'L>0'THEN'I:=ITAG(/L/)'ELSE'I:=N+1;'GOTO'BRCH'END';
SUS:'IF'IP=K'THEN'DE(/K/):=DE(/K/)-CF*CE(/LI/);BRCH:LI:=LNXT(/LI/);
'IF'LI>0'THEN''BEGIN'IP:=ITAG(/LI/);'GOTO'REX'END';LK:=LNXT(/LK/);
'IF'LK>0'THEN''GOTO'REY;FNJ:'END''END'REDUCT;

'PROCEDURE'SOLVE(V,N,N1,DE,LCOL,NOZE,NSEQ,BB,RP,TRANSP);
'VALUE'N,N1,TRANSP;'INTEGER'N,N1;'BOOLEAN'TRANSP;'ARRAY'V,DE;
'INTEGER''ARRAY'LCOL,NOZE,NSEQ,BB,RP;'BEGIN''IF'TRANSP'THEN''BEGIN'
'FOR'I:=N+1'STEP'1'UNTIL'N1'DO''BEGIN'II:=RP(/I/);JJ:=BB(/I/);
V(/I/):=V(/I/)/ELA(II,JJ)'END';'FOR'I:=1'STEP'1'UNTIL'N'DO''FOR'K:=N+1
'STEP'1'UNTIL'N1'DO''BEGIN'JJ:=RP(/K/);II:=BB(/I/);'IF'ELA(JJ,II)'NE'0
```

```
'THEN'V(/I/):=V(/I/)-ELA(JJ,II)*V(/K/)'END''END';'FOR'J:=1'STEP'1'UNTIL'
N'DO''BEGIN'K:=NSEQ(/J/);CF:=DE(/K/)*V(/K/);V(/K/):=CF;L:=LCOL(/K/);
REA:'IF'L<=0'THEN''GOTO'FJ1;I:=ITAG(/L/);V(/I/):=V(/I/)-('IF'TRANSP
'THEN'RE(/L/)/DE(/K/)'ELSE'CE(/L/))*CF;L:=LNXT(/L/);'GOTO'REA;FJ1:'END';
'FOR'J:=N-1'STEP'-1'UNTIL'1'DO''BEGIN'K:=NSEQ(/J/);SUM:=V(/K/);
L:=LCOL(/K/);REB:'IF'L>0'THEN''BEGIN'I:=ITAG(/L/);SUM:=SUM-('IF'
TRANSP'THEN'CE(/L/)*DE(/K/)'ELSE'RE(/L/))*V(/I/);L:=LNXT(/L/);'GOTO'REB
'END';V(/K/):=SUM'END';'IF''NOT'TRANSP'THEN''BEGIN''FOR'I:=N+1'STEP'1
'UNTIL'N1'DO''BEGIN'II:=RP(/I/);'FOR'K:=1'STEP'1'UNTIL'N'DO''BEGIN'
JJ:=BB(/K/);'IF'ELA(II,JJ)'NE'0'THEN'V(/I/):=V(/I/)-ELA(II,JJ)*V(/K/)
'END';V(/I/):=V(/I/)/ELA(II,BB(/I/))'END''END''END'SOLVE;

'PROCEDURE'SYST(X,NS,NB1,NB2,DE,LCOL,NOZE,NSEQ,BB,RP,
TRANSP);'VALUE'NS,NB1,NB2,TRANSP;'INTEGER'NS,NB1,NB2;
'BOOLEAN'TRANSP;'ARRAY'X,DE;'INTEGER''ARRAY'LCOL,NOZE,NSEQ,BB,RP;
'BEGIN''REAL'U;'ARRAY'V(/1:NB1/),BL3(/NB1+1:NB2,NB1+1:NB2/);
'FOR'KP:=NB1+1'STEP'1'UNTIL'NB2'DO''BEGIN''IF'TRANSP'THEN''BEGIN'
'FOR'IP:=1'STEP'1'UNTIL'NB1'DO'V(/IP/):=ELA(RP(/KP/),BB(/IP/))'END'
'ELSE''FOR'IP:=1'STEP'1'UNTIL'NB1'DO'V(/IP/):=ELA(RP(/IP/),BB(/KP/));
SOLVE(V,NS,NB1,DE,LCOL,NOZE,NSEQ,BB,RP,TRANSP);
'FOR'IP:=NB1+1'STEP'1'UNTIL'NB2'DO''BEGIN''IF'TRANSP'THEN'
BL3(/IP,KP/):=-ELA(RP(/KP/),BB(/IP/))'ELSE'BL3(/IP,KP/):=-ELA(RP(/IP/),
BB(/KP/));'IF'TRANSP'THEN''BEGIN''FOR'LP:=1'STEP'1'UNTIL'NB1
'DO'BL3(/IP,KP/):=BL3(/IP,KP/)+ELA(RP(/LP/),BB(/IP/))*V(/LP/)'END'
'ELSE''FOR'LP:=1'STEP'1'UNTIL'NB1'DO'BL3(/IP,KP/):=BL3(/IP,KP/)+ELA(RP(/
IP/),BB(/LP/))*V(/LP/)'END''END';'FOR'IP:=1'STEP'1'UNTIL'NB1'DO'V(/IP/):
=X(/IP/);SOLVE(V,NS,NB1,DE,LCOL,NOZE,NSEQ,BB,RP,TRANSP);
'FOR'IP:=NB1+1'STEP'1'UNTIL'NB2'DO''BEGIN'X(/IP/):=-X(/IP/);'IF'TRANSP
'THEN''BEGIN''FOR'LP:=1'STEP'1'UNTIL'NB1'DO'X(/IP/):= X(/IP/)+ELA(RP(/
LP/),BB(/IP/))*V(/LP/)'END''ELSE''FOR'LP:=1'STEP'1'UNTIL'NB1'DO'X(/IP/):
=X(/IP/)+ELA(RP(/IP/),BB(/LP/))*V(/LP/)'END';'FOR'KP:=NB1+1'STEP'1'UNTIL
'NB2'DO''BEGIN'MAX:=ABS(BL3(/KP,KP/));LP:=KP;'FOR'IP:=KP+1'STEP'1
'UNTIL'NB2'DO''BEGIN'U:=ABS(BL3(/IP,KP/));'IF'U>MAX'THEN''BEGIN'MAX:=U;
LP:=IP'END''END';'IF'LP'NE'KP'THEN''BEGIN''FOR'IP:=KP'STEP'1'UNTIL'NB2
'DO''BEGIN'U:=BL3(/KP,IP/);BL3(/KP,IP/):=BL3(/LP,IP/);BL3(/LP,IP/):=U
'END';U:=X(/KP/);X(/KP/):=X(/LP/);X(/LP/):=U'END';U:=BL3(/KP,KP/);
'FOR'LP:=KP+1'STEP'1'UNTIL'NB2'DO''BEGIN'BL3(/KP,LP/):=BL3(/KP,LP/)/U;
'FOR'IP:=NB1+1'STEP'1'UNTIL'NB2'DO''IF'IP'NE'KP'THEN'BL3(/IP,LP/):=
BL3(/IP,LP/)-BL3(/IP,KP/)*BL3(/KP,LP/)'END';X(/KP/):=X(/KP/)/U;
'FOR'IP:=NB1+1'STEP'1'UNTIL'NB2'DO''IF'IP'NE'KP'THEN'X(/IP/):=X(/IP/)-
BL3(/IP,KP/)*X(/KP/)'END';'IF'TRANSP'THEN''BEGIN''FOR'IP:=1'STEP'1'UNTI
L'NB1'DO''FOR'KP:=NB1+1'STEP'1'UNTIL'NB2'DO'X(/IP/):=X(/IP/)-ELA(RP(/KP/
),BB(/IP/))*X(/KP/)'END''ELSE''FOR'IP:=1'STEP'1'UNTIL'NB1'DO''FOR'KP:=NB
1+1'STEP'1'UNTIL'NB2'DO'X(/IP/):=X(/IP/)-ELA(RP(/IP/),BB(/KP/))*X(/KP/);
SOLVE(X,NS,NB1,DE,LCOL,NOZE,NSEQ,BB,RP,TRANSP)'END'SYST;

'PROCEDURE'SIMPB(BM,BP,BB,RP,M,NP,NB2,DELTA,ERR);
'VALUE'M,NP,NB2,DELTA;'INTEGER'M,NP,NB2;'REAL'DELTA;'INTEGER'
'ARRAY'BB,RP;'BOOLEAN''ARRAY'BM,BP;'LABEL'ERR;'BEGIN''ARRAY'SOL,UU(/1:M
/);'INTEGER''ARRAY'BB1(/1:M/);'BOOLEAN''ARRAY'MF(/1:M/);CBS:'BEGIN''ARRA
Y'DE(/1:NS/);'INTEGER''ARRAY'LCOL,NOZE,NSEQ (/1:NS/);REDUCT(NS,DE,
LCOL,NOZE,NSEQ,BB,RP,ERR);CSOL:'FOR'I:=1'STEP'1'UNTIL'M'DO'
'BEGIN'II:=RP(/I/);U:=A(/II/);'FOR'J:=1'STEP'1'UNTIL'NN'DO''BEGIN'
'IF'BP(/J/)'THEN'U:=U-ELA(II,J)*X1(/J/)'ELSE''IF'BM(/J/)'THEN'U:=U-
ELA(II,J)*XO(/J/)'END';SOL(/I/):=U'END';SYST(SOL,NS,NB1,M,DE,LCOL,NOZE,
```

```
NSEQ,BB,RP,'FALSE');'FOR'I:=1'STEP'1'UNTIL'M'DO''BEGIN'J:=BB(/I/);
'IF'J<=N'THEN''BEGIN;''IF'SOL(/I/)<XO(/J/)'THEN''BEGIN'VNRM(/J/):='TRUE';
VNRP(/J/):='FALSE''END''ELSE''IF'SOL(/I/)>X1(/J/)'THEN''BEGIN'VNRM(/J/)
:='FALSE';VNRP(/J/):='TRUE''END''ELSE''BEGIN'VNRM(/J/):='FALSE';VNRP(/J/
):='FALSE''END''END';ELSE''IF'SOL(/I/)<0'THEN''BEGIN'VNRM(/J/):='TRUE';
VNRP(/J/):='FALSE''END''ELSE''BEGIN'
VNRM(/J/):='FALSE';VNRP(/J/):='FALSE''END''END';VNRM(/NN/):=VNRP(/NN/):=
'FALSE';CCC:'FOR'I:=1'STEP'1'UNTIL'M'DO''BEGIN'J:=BB(/I/);'IF'VNRM(/J/)'
THEN'UU(/I/):=DELTA'ELSE''IF'VNRP(/J/)'THEN'UU(/I/):=-DELTA'ELSE'UU(/I/)
:=0;''IF'J=NN'THEN'UU(/I/):=1'END';SYST(UU,NS,NB1,M,DE,LCOL,NOZE,
NSEQ,BB,RP,'TRUE');SP:=SM:=0;UM:=EP1;UP:=EP2;'FOR'I:=1
'STEP'1'UNTIL'NP'DO';'IF'BM(/I/)'OR'BP(/I/)'THEN''BEGIN'U:=0;
'FOR'J:=1'STEP'1'UNTIL'M'DO'U:=U-UU(/J/)*ELA(RP(/J/),I);'IF'BM(/I/)
'THEN''BEGIN''IF'U>UM'THEN''BEGIN'SM:=I;UM:=U'END''END''ELSE''IF'U<UP
'THEN''BEGIN'SP:=I;UP:=U'END''END';''IF'SM=0'AND'SP=0'THEN';GOTO'OPT;
'IF'UM>=-UP'THEN''BEGIN'S:=SM;BOOL:='TRUE''END''ELSE''BEGIN'S:=SP;
BOOL:='FALSE''END';U:='IF'S<=N'THEN'X1(/S/)-XO(/S/)'ELSE'UPUI;I1:=3;
'FOR'I:=1'STEP'1'UNTIL'M'DO'UU(/I/):=ELA(RP(/I/),S);SYST(UU,NS,NB1,M,DE,
LCOL,NOZE,NSEQ,BB,RP,'FALSE');R:=0;'FOR'I:=1'STEP'1'UNTIL'M'DO'
'BEGIN'T:=UU(/I/);'IF'BOOL'THEN'T:=-T;J:=BB(/I/);U1:=-SOL(/I/);
I2:=0;'IF'T<EP3'THEN''BEGIN'I2:=1;''IF'VNRM(/J/)'THEN'I2:=0'ELSE'
'IF'J<=N'THEN'U1:=XO(/J/)+U1'ELSE''IF'J=NN'THEN'I2:=0'END''ELSE''IF'T>E
P4'THEN''BEGIN''IF'VNRP(/J/)'THEN'I2:=0'ELSE''IF'J<=N'THEN''BEGIN'I2:=2;
U1:=X1(/J/)+U1'END''ELSE'I2:=0'END';'IF'I2<>0'THEN''BEGIN'U1:=U1/T;
'IF'U>U1'THEN''BEGIN'U:=U1;R:=I;I1:=I2'END''END''END';'IF'I1=3
'THEN''BEGIN'BP(/S/):=BOOL;BM(/S/):='NOT'BOOL'END''ELSE''BEGIN'BP(/S/):
=BM(/S/):='FALSE';'IF'I1=2'THEN'BP(/BB(/R/)/):='TRUE''ELSE'BM(/BB(/R/)/)
:='TRUE''END';
NS1:=NS;'FOR'I:=1'STEP'1'UNTIL'M'DO'BB1(/I/):=BB(/I/);'FOR'I:=1'STEP'1
'UNTIL'M'DO'MF(/I/):='TRUE';NS:=0;'FOR'J:=1'STEP'1'UNTIL'2*NE-1'DO';'IF'
'NOT'BP(/J/)'AND''NOT'BM(/J/)'THEN''BEGIN'NS:=NS+1;BB(/NS/):=J;RP(/NS/)
:=J+1;MF(/J+1/):='FALSE''END';NB1:=NS;K:=M+1;'FOR'J:=2*NE'STEP'1'UNTIL'
NP'DO';'IF';'NOT'BP(/J/)'AND';'NOT'BM(/J/)'THEN';'BEGIN';'IF'J=NN'THEN';'GOTO'
DER;'FOR'I:=1'STEP'1'UNTIL'NS'DO';'IF'ELA(RP(/I/),J)<>0'THEN';'GOTO'DER;
'FOR'I:=2'STEP'1'UNTIL'NB2'DO';'IF'MF(/I/)'AND'ELA(I,J)<>0'THEN';'GOTO'
TDS;'GOTO'DER;TDS:NB1:=NB1+1;BB(/NB1/):=J;RP(/NB1/):=I;
MF(/I/):='FALSE';'GOTO'TD1;DER:K:=K-1;BB(/K/):=J;TD1:'END';
K:=NB1;'FOR'I:=1'STEP'1'UNTIL'M'DO';'IF'MF(/I/)'THEN';'BEGIN'K:=K+1;
RP(/K/):=I'END';'IF'NS<>NS1'THEN';'GOTO'CBS;'FOR'I:=1'STEP'1'UNTIL'NS'
DO';'IF'BB(/I/)<>BB1(/I/)'THEN';'GOTO'CBS;'GOTO'CSOL;OPT:'FOR'I:=1'STEP'
1'UNTIL'M'DO';'BEGIN'J:=BB(/I/);'IF'VNRM(/J/)'OR'VNRP(/J/)'THEN';'GOTO'AUG
D'END';'GOTO'FSVB;FRM4:'FORMAT'(18H DELTA TROP FAIBLE);AUGD:
'WRITE'(108,FRM4);DELTA:=10*DELTA;'GOTO'CCC'END';FSVB:'FOR'I:=1
'STEP'1'UNTIL'N'DO';'BEGIN';'IF'BP(/I/)'THEN'Z(/I/):=X1(/I/)'ELSE';'IF'BM(
/I/)'THEN'Z(/I/):=
XO(/I/)'END';'FOR'I:=1'STEP'1'UNTIL'M'DO';'BEGIN'R:=BB(/I/);'IF'R<=NN'THE
N'Z(/R/):=SOL(/I/)'END''END'SIMPB;

FRM5:'FORMAT'(16F5.2);'READ'(105,FRM5)'FOR'I:=1'STEP'1'UNTIL'F1'DO'
(GD(/I/));FRM6:'FORMAT'(F8.2,F7.2,I2,L2);'READ'(105,FRM6)'FOR'
I:=1'STEP'1'UNTIL'NE'DO'(C1(/I/),D1(/I/),P1(/I/),Q2(/I/));FRM7:'FORMAT'
(2I2,3F6.4,F7.4);'FOR'K:=1'STEP'1'UNTIL'NLN'DO';'BEGIN';'READ'(105,FRM7)
RG1(/K/),RG2(/K/),U,U1,T,T1,B1(/K/):=U1;B2(/K/):=T1;CS1(/K/):=
U*U1*COS(T1-T);SN1(/K/):=U*U1*SIN(T1-T)'END';NS:=NB1:=
2*NE-1;X1(/NN/):='E'5;XO(/NN/):=-'E'5;ICEN:=0;'FOR'I:=1'STEP'1
```

```
'UNTIL'NE'D0''BEGIN'X0(/I/):=205;X1(/I/):=240;YD(/I/):=205'END';
'FOR'I:=NE+1'STEP'1'UNTIL'2*NE-1'DO''BEGIN'X0(/I/):=-.5;X1(/I/):=.5;
YD(/I/):=-.5'END';FRM8:'FORMAT'(2I4);'READ'(105,FRM8)'FOR'I:=2*NE
'STEP'1'UNTIL'N'DO'(X0(/I/),X1(/I/));'FOR'I:=2*NE'STEP'1'UNTIL'N'DO'YD(/
I/):=X1(/I/);'FOR'I:=1'STEP'1'UNTIL'2*NE'DO''BEGIN'BB(/I/):=I;RP(/I/):=I
+1;BM(/I/):=BP(/I/):='FALSE''END';'FOR'I:=2*NE+1'STEP'1'UNTIL'NP+ADC'DO'
'BEGIN'BM(/I/):='TRUE';BP(/I/):='FALSE''END';BB(/M1/):=NN;RP(/M1/):=1;
BM(/NN/):='FALSE';RET:ICEN:=ICEN+1;FRM9:'FORMAT'(E10.0);
'IF'ICEN=2'THEN''READ'(105,FRM9)KKK;ILIN:=0;'FOR'I:=1'STEP'1
'UNTIL'N'DO'X2(/I/):=YD(/I/);FRM10:'FORMAT'(6HOC MAX);'IF'ICEN>
ITMAX'THEN''BEGIN''WRITE'(108,FRM10);'GOTO'FNI'END';RET1:'FOR'I:=1
'STEP'1'UNTIL'BSAC'DO''BEGIN'AC(/I/):=0;TLI(/I/):=ICO(/I/):=CH(/I/):=0
'END';'FOR'I:=1'STEP'1'UNTIL'N'DO'TC(/I/):=0;F:=1;LR1:=1;
CTR:=0;CONT(2,FDIST,YD);ILIN:=ILIN+1;'IF'ILIN=1'THEN'
'BEGIN'FCT3:=FCT1;FCT1:=VCONT(/1/);'IF'ABS((FCT3-FCT1)/
FCT1)<PREC'THEN''GOTO'FNI'END';FCT2:=VCONT(/1/);
FRM11;'FORMAT'(1HO,I8,E23.12);'WRITE'(108,FRM11)ICEN,FCT2;
LR:=0;'FOR'I:=1'STEP'1'UNTIL'M1'DO''BEGIN'U:=0;J:=LR;'FOR'J:=J+1'WHILE'
ILI(/J/)=I'DO'U:=U+AC(/J/)*YD(/ICO(/J/)/);U:='IF'I=1'THEN'U+VCONT(/1/)-
FCT1'ELSE'U+VCONT(/I/);AA:=NOR(/I/);A(/I/):=U*AA;J:=LR;'FOR'J:=J+1
'WHILE'ILI(/J/)=I'DO''BEGIN'AC(/J/):=AC(/J/)*AA;LR:=LR+1'END''END';I:=0;
'FOR'I:=I+1'WHILE'ILI(/I/)=1'DO'AC(/I/):=KKK*AC(/I/);A(/1/):=KKK*A(/1/);
SIMPB(BM,BP,BB,RP,M1,NP,M1,
DELTA,IMP);'IF'ICEN<ICEN'THEN''GOTO'E10;MM2:=M1;NP1:=NP;
NS2:=NS;NB12:=NB1;'FOR'I:=1'STEP'1'UNTIL'M1+ADC'DO''BEGIN'
BB2(/I/):=BB(/I/);RP2(/I/):=RP(/I/)'END';'FOR'I:=1'STEP'1'UNTIL'NP+ADC'D
O''BEGIN'BM2(/I/):=BM(/I/);BP2(/I/):=BP(/I/)'END';FCT5:=UPUI;
BIDULE:=0;E11:RZMAX(7,X2,YD);MM2:=MM2+1;NP1:=NP1+1;
'IF'MM2>M1+ADC'THEN''BEGIN''FOR'I:=1'STEP'1'UNTIL'N'DO'YD(/I/):=X5(/I/);
NS:=NS2;NB1:=NB12;'GOTO'ETTT'END';CONT(1,FDIST,X4);
U:=UPUI;'FOR'I:=1'STEP'1'UNTIL'M1'DO''BEGIN'U1:=ABS(VCONT(/I/));
'IF'U1<U'THEN''BEGIN'U:=U1;CTR:=I'END''END';F:=MM2;LR:=LR1-1;
CONT(2,FDIST,X4);U:=0;J:=LR;'FOR'J:=J+1'WHILE'ILI(/J/)=MM2'DO'U:=U+
AC(/J/)*X4(/ICO(/J/)/);U:='IF'CTR=1'THEN'U+VCONT(/1/)-FCT1'ELSE'U+
VCONT(/CTR/);AA:=NOR(/MM2/);A(/MM2/):=U*AA;J:=LR;'FOR'J:=J+1'WHILE'
ILI(/J/)=MM2'DO'AC(/J/):=AC(/J/)*AA;
BM2(/NP1/):=BP2(/NP1/):='FALSE';BB2(/MM2/):=NP1;
RP2(/MM2/):=MM2;SIMPB(BM2,BP2,BB2,RP2,MM2,NP1,M1,DELTA,IMP);
'GOTO'E11;E10:RZMAX(7,YD,YD);ETTT:'IF'ICEN=1'AND''NOT'BOOL1
'THEN''GOTO'RET1;'IF'ILIN<ILINMA'THEN''GOTO'RET1'ELSE''GOTO'RET;
FNI:CONT(3,FDIST,X2);FRM12:'FORMAT'(10HOOPTIMUM  ,E18.12);
'WRITE'(108,FRM12)VCONT(/1/);FRM13:'FORMAT'(12HOCONTRAINTES);
'WRITE'(108,FRM13);FRM14:'FORMAT'(1HO,7E18.12);'WRITE'(108,FRM14)'FOR'
I:=2'STEP'1'UNTIL'M1'DO'(VCONT(/I/));FRM15:'FORMAT'(9HOSOLUTION);
'WRITE'(108,FRM15);FRM16:'FORMAT'(1HO,7F19.12);'WRITE'(108,FRM16)'FOR'
I:=1'STEP'1'UNTIL'N'DO'(X2(/I/));FRM17:'FORMAT'(24HOETAT PHYSIQUE DU RES
EAU);'WRITE'(108,FRM17);FRM18:'FORMAT'(6HONOEUD,8X,1HU,13X,5HTHETA,13X,
1HP,15X,1HQ);'WRITE'(108,FRM18);J:=2*NE;I:=2*NE+F1;'FOR'I:=1'STEP'1'UNTI
L'NE'DO''BEGIN'U:='IF'I'NE'NE'THEN'X2(/I+NE/)'ELSE'0;U1:=0;
'FOR'K:=1'STEP'1'UNTIL'P1(/I/)'DO''BEGIN'U1:=U1+X2(/J/);J:=J+1'END';
'IF'Q2(/I/)'THEN''BEGIN'T:=X2(/L/)-VCONT(/I/);L:=L+1'END''ELSE'T:=0;
FRM19;'FORMAT'(1H ,I8,F16.8,F16.12,F16.8,F16.8);
'WRITE'(108,FRM19)I,X2(/I/),U,U1,T'END''END';IMP:'END'
```

CONCLUSION

On remarque d'abord l'importance de la modélisation des problèmes considérés. Les essais numériques montrent que le modèle doit autant que possible vérifier les hypothèses sous lesquelles la convergence théorique de la méthode des centres linéarisée est assurée (en particulier, il est important que la fonction économique soit continuement différentiable). De plus, les problèmes comportant des contraintes d'égalités peuvent être pris en compte, au prix d'une transformation simple qui conduit à un modèle dont le domaine possède un intérieur non vide.

Pour résoudre le problème, il faut ensuite ajuster le coefficient de pondération k de la fonction économique. Si celui-ci est trop faible, le rapport de progression de la valeur de la fonction économique d'une itération à l'autre reste insuffisant ; s'il est trop fort, on a des ennuis numériques dus à la trop grande disparité des coefficients de la matrice des contraintes linéaires. Pour ajuster k, il suffit d'effectuer quelques essais successifs sur un petit nombre de troncatures, en donnant au coefficient des valeurs choisies à un facteur 10 près, et sans ajouter de contraintes de centrage. La valeur optimale de k semblant dépendre davantage de la forme du problème que de sa taille, on peut penser que pour un type de problème donné cette valeur est une constante (pour le Dispatching Economique, cette valeur est $k = 10^{-5}$).

Le second paramètre important est le nombre de contraintes additionnelles de centrage. Si ce nombre est trop faible, le nombre de troncatures nécessaires pour obtenir l'optimum risque d'être élevé ; s'il est trop fort, le volume des calculs par itération devient trop important. Sa valeur optimale dépend de la taille du problème traité et doit également être déterminée par essais successifs ; toutefois la différence entre le nombre de variables et le nombre de contraintes en donne l'ordre de grandeur. On remarque de plus qu'il est préférable d'utiliser la variante de l'algorithme de centrage introduite en I.5.3.

Le troisième paramètre, la précision de la recherche du maximum de la F-distance sur un segment, influe beaucoup moins sur la convergence pratique de la méthode. La valeur de dic peut être fixée à priori à une valeur suffisamment grande (en général dic = 10^9 donne de bons résultats).

L'utilisation du creux des problèmes, en particulier dans la méthode simpliciale, permet d'optimiser l'encombrement mémoire et donc de traiter des problèmes de grandes tailles.

Sur les problèmes traités, on constate qu'on obtient une excellente précision sur la solution en un nombre de troncatures en général compris entre 10 et 15.

Enfin, des procédés de maximisation de la F-distance sur un segment tirant parti de la forme de cette fonction, ainsi que l'exploitation du fait que, dans la méthode simpliciale, la base optimale des programmes linéaires varie peu d'une linéarisation à l'autre, permettent un gain de temps de calcul.

BIBLIOGRAPHIE

[1] BARTELS (R.H.), GOLUB (G.H.)

The simplex method of linear programming using LU decomposition
Communications of the ACM
vol 12 n°5, mai 1969, p 266

[2] BEUNEU (J.)

Adaptation de la méthode des centres linéarisée à un problème de Dispatching Economique
Université des Sciences et Techniques de Lille
Octobre 1972.

[3] BLANCHON (G.), DAUPHIN (G.), FEINGOLD (D.), SPOHN (D.)

Le problème du Dispatching Economique : méthode mathématique et résultats
Note EDF n° HR 9018/1, janvier 1969

[4] BRAYTON (R.K.), GUSTAVSON (F.G.), WILLOUGHBY (R.A.)

Some results on sparse matrices
Mathematics of Computation
vol 24, n°112, octobre 1970, p 937-954

[5] BUI TRONG LIEU, HUARD (P.)

La méthode des centres dans un espace topologique
Numerische Mathematik
vol 8, 1966, p 56.

[6] CARPENTIER (J.)

Contribution à l'étude du Dispatching Economique
Rapport EDF n° HX 583/394, février 1962

[7] CARPENTIER (J.)

Eliminations ordonnées, un processus diminuant le volume des calculs dans la résolution des systèmes linéaires à matrice creuse
3e congrès de calcul et de traitement de l'information, mai 1963, Dunod.

[8] COLVILLE (A.R.)

Acomparative study on non-linear programming codes
IBM New-York Scientific Center, Technical Report n° 320-2949, juin 1968.

[9] DANTZIG (G.B.)

Compact basis triangularization for the simplex method
Recent advances in mathematical programming (R.L. Graves, P. Wolfe),
Mc Graw Hill Book Company, p.125

[10] DENEL (J.)

Résolution de problèmes d'optimisation non linéaires par la méthode des centres linéarisée
Thèse de doctorat de spécialité,
Université des Sciences et Techniques de Lille
juin 1972.

[11] DENEL (J.)

Résolution de problèmes d'optimisation non linéaires par la méthode des centres linéarisée
Bulletin de la direction des études et recherches
EDF, série C n°1 1973, p 5-42.

[12] DENEL (J.), HUARD (P.)

Programmation non linéaire et linéarisation
Applications of optimization methods for large-scale resource-allocation problems,
Nato conference, Elsinore, Denmark, juillet 1971.

[13] BODU (J.C.), MERLIN (A.)

Une application de la programmation linéaire à l'étude des réseaux électriques de grande taille
Bulletin de la direction des études et recherches, EDF, série C n°1, 1974, p29-56

[14], FIACCO (A.V.)

Second order sufficient conditions for weak and strict constrained minima
Siam J.Appl Math. vol 16, n°1, 1968, p 105-108

[15] FORREST (J. J.H.), TOMLIN (J.A.)

Updated triangular factors of the basis to maintain sparsity in the product form simplex method
Mathematical Programming 2, 1972, p 263-278

[16] GEOFFRION (A. M.)

Elements of large-scale mathematical programming
Management Science, vol 16, n°11, juillet 1970
p 652-691

[17] GOLDFARB (D.)

Modification methods for inverting matrices and solving systems of linear algebraic equations
IBM, technical report n°320-2998, janvier 1971

[18] GUIGNARD (M.)

Generalized Kuhn-Tucker conditions for mathematical programming problems in a Banach space
Siam J;Control, vol 7, n°2, 1969, p 232-241

[19] HELD (M.), WOLFE (P.), CROWDER (H.P.)

Validation of subgradient optimization
Mathematical Programming,vol 6, n°1, février 1974 p 62-88

[20] HELLERMAN (E.), RARICK (D.)

Reinversion with the preassigned pivot procedure
Mathematical Programming 1,1971, p 195-216

[21] HUARD (P.)

Resolution of mathematical programming problems with non linear constraints by the method of centres
Non linear programming, ed. Abadie, North-Holland Publications, Amsterdam, 1967, p 206-219

[22] HUARD (P.)

Programmation mathématique convexe
Bulletin de la direction des études et recherches, EDF, série C n°1, 1968, p 61-74

[23] HUARD (P.)

Méthode des centres et méthode des centres par majorations
Bulletin de la direction des études et recherches, EDF, série C n°2, 1970, p33-52

[24] HUARD (P.)

Tour d'horizon en programmation non linéaire
Bulletin de la direction des études et recherches, EDF, série C, n°1, 1971, p 35-70

[25] KOWALIK (J.), OSBORNE (M. R.)

Methods for unconstrained optimization problems
Elsevier, New-York, 1968

[26] MC CORMICK (G. P.)

Second order conditions for constrained minima
Siam J. Appl. Math. vol 15, n°3, 1967, p 641-652

[27] MANGASARIAN (O. L.)

Nonlinear programming
Mc Graw Hill Series in Syst. Science,
Mc Graw Hill Book Company, 1969

[28] MERZLYAKOV (Y. I.)

On a relaxation method of solving systems of linear inequalities
USSR Computational Mathematics, 1963, p 504-510

[29] MOTZKIN (T. S.), SCHOENBERG (I. J.)

The relaxation method for linear inequalities
Canad. J. Math, 6 n°3, 1954, p 393-404

[30] POOCH (U. W), NIEDER (A.)

A survey of indexing techniques for sparse matrices
ACM Computing Surveys, vol 5, n°2, juin 1973, p 109-133.

[31] REID (J. K.)

Large sparse sets of linear equations
Academic Press, 1971

[32] TEWARSON (R. P.)

Computations with sparse matrices
Siam Review, vol 12, n°4, octobre 1970, p 527-543

[33] TEWARSON (R. P.)

On the product form of inverses of sparse matrices
Siam Review, vol 8, n°3, juillet 1966, p 336-342

[34] TEWARSON (R. P.)

The Crout reduction for sparse matrices
Computer Journal 12, 1969, p 158-159

[35] TOMLIN (J. A.)

Maintaining a sparse inverse in the simplex method
Operations Research Department, Stanford University
IBM J. Res. Develop., vol 16, n°4, juillet 1972,
p 415-423

[36] WILKINSON (J. H.)

The algebraic eigenvalue problem
Oxford University Press, London, 1965

[37] WOLFE (P.)

The composite simplex algorithm
SIAM Review, vol 7, n°1, janvier 1965, p 42-54

[38] ZOLLENKOPF (K.)

Bi-factorisation, basic computational algorithm and programming techniques

dans REID [31], p 75-96