

THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES

Méthode des multiplicateurs et application à des problèmes non différentiables

Dropsy, Christiane

Award date:
1977

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

METHODE DES MULTIPLICATEURS
ET
APPLICATION A DES PROBLEMES
NON DIFFERENTIABLES

DROPSY
CHRISTIANE

FMB1/1977/12



175984
L955 3452741

Je tiens tout d'abord à remercier Monsieur
V.Hien NGUYEN ainsi que Monsieur J.J. STODIOT
pour leur aide et leurs précieux conseils.

INTRODUCTION

Considérons le problème de programmation non linéaire :

$$\begin{aligned} \min f(x) \\ g(x) \leq 0 \end{aligned}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$

qui consiste à trouver un point \bar{x} tel que $g(\bar{x}) \leq 0$ et $f(\bar{x}) \leq f(x)$ pour tout x satisfaisant $g(x) \leq 0$.

Pour résoudre ce problème , on connaît bien la méthode de pénalisation quadratique [6], qui consiste à minimiser la fonction de pénalisation :

$$f(x) + \sum_{i=1}^m \frac{\alpha_k}{2} [\max (0 , g_i(x))]^2$$

pour une suite de nombres α_k strictement positifs et tendant vers l'infini.

Cette méthode présente un inconvénient : lorsque α_k prend des valeurs très grandes, la matrice Hessienne de la fonction de pénalisation devient mal conditionnée et il devient de plus en plus difficile de la minimiser.

Pour éviter cette difficulté , Rockafellar [15] en 1970 d'une part et Powell et Hestenes [14,7] en 1969 d'autre part ont proposé une autre méthode que nous nous proposons d'étudier dans une première partie. Cette méthode est appelée méthode des multiplicateurs, mais elle est connue aussi sous le nom de méthode du Lagrangien augmenté ou méthode du lagrangien pénalisé (nous verrons pourquoi quand nous la décrirons.) L'exposé de cette première partie est basé sur un article de P.Bertsekas [1] .

Ensuite , dans la deuxième partie , nous verrons com-

ment cette méthode va donner lieu à une méthode pour résoudre certains problèmes non différentiables , c'est-à-dire des problèmes où la fonction $f(.)$ du problème ci-dessus est non-différentiable. L'exposé de cette seconde partie sera également basé sur un article de P.Bertsekas[2]

Enfin, dans la troisième partie , nous montrerons comment nous avons implémenté la méthode des multipliateurs et nous donnerons quelques résultats.

PREMIERE PARTIE

LA METHODE DES MULTIPLICATEURS

Introduction

a. Description de la méthode et présentation de l'analyse

Nous allons étudier dans cette première partie une méthode numérique pour résoudre le problème suivant de minimisation avec contraintes :

$$\begin{aligned} & \text{minimiser } f(x) \\ & \text{sous contraintes : } h_i(x) = 0 \quad i=1, \dots, m \end{aligned} \quad (1)$$

où $f: \mathbb{R}^n \rightarrow \mathbb{R}$ et $h_i: \mathbb{R}^n \rightarrow \mathbb{R}$ sont des fonctions données.

Si \bar{x} est une solution optimale pour ce problème et $\bar{y} = (\bar{y}^1, \dots, \bar{y}^m)$ le vecteur optimal des multiplicateurs de Lagrange associé, alors le lagrangien du problème :

$$L_0(x, y) = f(x) + \sum_{i=1}^m y^i h_i(x)$$

$$L_0 : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$$

satisfait au point (\bar{x}, \bar{y}) les conditions nécessaires d'optimalité de Kuhn-Tucker, et en particulier :

$$\nabla L_0(\bar{x}, \bar{y}) = \nabla f(\bar{x}) + \sum_{i=1}^m \bar{y}^i \nabla h_i(\bar{x}) = 0 \quad (2)$$

où le symbole ∇ indique que dans le gradient, on dérive par rapport à x . (Pour le détail, voir par exemple [6], Fiacco et Mc Cormick (1968)).

Mais malheureusement \bar{x} n'est pas toujours un minimum de $L(\cdot, \bar{y})$ sur \mathbb{R}^n et d'ailleurs, on ne connaît pas \bar{y} .

En effet, on aimerait bien ramener ce problème avec contraintes à un problème sans contraintes pour lequel on connaît beaucoup de méthodes efficaces de minimisation. (Voir par exemple [9], Murray (1972)).

Une idée consiste à minimiser par rapport à $x \in \mathbb{R}^n$, la fonction :

$$f(x) + \sum_{i=1}^m y^i h_i(x) + \frac{c_k}{2} \sum_{i=1}^m [h_i(x)]^2 \quad (3)$$

pour une suite de vecteurs $y_k = (y_k^1, \dots, y_k^m)$ et une suite de scalaires $\{c_k\}$. Cette méthode est souvent connue sous le nom de méthode des multiplicateurs.

La fonction (3) peut être interprétée comme une fonction lagrangienne $[f(x) + \sum_{i=1}^m y_k^i h_i(x)]$ à laquelle on a ajouté un terme de pénalisation $[\frac{c_k}{2} \sum_{i=1}^m [h_i(x)]^2]$. C'est la raison pour laquelle on l'appelle "lagrangien augmenté".

On a proposé plusieurs manières d'ajuster les scalaires c_k : l'une d'elles consiste à faire croître c_k jusqu'à l'infini d'une manière bien déterminée, par exemple en posant :

$$c_k = r^k \quad \text{où } r > 1$$

Une autre possibilité est de garder c_k fixé après un certain indice.

Après chaque minimisation sans contraintes, donnant le point de minimum x_k , le vecteur y_k est ajusté par l'itération suivante :

$$y_{k+1}^i = y_k^i + c_k h_i(x) \quad i=1, \dots, m \quad (4)$$

à condition que $y_k^i + c_k h_i(x) \in S$, un ensemble borné

donné arbitraire ;

et où y_k^i , y_{k+1}^i désignent la $i^{\text{ème}}$ composante respectivement de y_k , y_{k+1} .

Nous montrerons la convergence de cette itération vers \bar{y} , le multiplicateur optimal de Lagrange du problème (1).

REMARQUE

Si pour le lagrangien augmenté

$$L(x, y_k, c_k) = f(x) + \sum_{i=1}^m y_k^i h_i(x) + \sum_{i=1}^m [h_i(x)]^2 \quad (5)$$

après chaque minimisation sans contrainte par rapport à x , au lieu de mettre à jour les y_k^i par l'itération (4), on garde $y_k^i = 0$ pour tout k et $i=1, \dots, m$ et que la suite de scalaires $\{c_k\}$ tend vers l'infini, on retombe sur la méthode de pénalisation quadratique qui consiste à minimiser la fonction :

$$f(x) + \frac{c_k}{2} \sum_{i=1}^m [h_i(x)]^2 \quad (6)$$

pour une suite non bornée de scalaires $\{c_k\}$.

Cette remarque nous servira dans la suite pour comparer les résultats de convergence de deux méthodes : celle des multiplicateurs que nous étudierons plus en détail et celle de pénalisation quadratique.

En effet, l'analyse que nous allons faire sera celle d'une méthode plus générale appelée méthode de pénalisation généralisée ; la méthode des multiplicateurs et la méthode de pénalisation quadratique en seront des cas particuliers. La méthode de pénalisation généralisée consiste en une minimisation du lagrangien augmenté pour une suite de scalaires $\{c_k\}$ et une suite

de vecteurs y_k à laquelle on impose uniquement d'être incluse dans S , un ensemble borné donné, arbitraire.

On pourra déduire les résultats des deux méthodes qui nous intéressent en prenant les cas particuliers :

1) pour la méthode des multiplicateurs

$$y_{k+1}^i = y_k^i + c_k h_i(x_k) \quad i=1, \dots, m$$

à condition que

$$y_k^i + c_k h_i(x_k) \in S$$

$$\text{sinon } y_{k+1}^i = y_k^i$$

2) pour la méthode de pénalisation quadratique

$$y_k = 0 \quad \text{pour tout } k$$

c_k tend vers l'infini quand k tend vers l'infini

L'analyse que nous allons faire comportera quatre parties :

A. Sous des hypothèses que nous spécifierons dans la suite, nous montrerons que pour la méthode de pénalisation générale décrite ci-dessus :

il existe des scalaires c^* et M positifs tels que pour tout c_k plus grand que c^* et y_k appartenant à S :

$$\|x_k - \bar{x}\| \leq M \frac{\|y_k - \bar{y}\|}{c_k} \quad (7)$$

$$\|y_{k+1} - \bar{y}\| \leq M \frac{\|y_k - \bar{y}\|}{c_k} \quad (8)$$

où \bar{x} , \bar{y} sont respectivement la solution optimale et le vecteur optimal des multiplicateurs de Lagrange pour le problème (1),

où x_k est un point minimisant localement le lagrangien augmenté (5) dans un voisinage de \bar{x} ,

y_{k+1} est donné en termes de c_k, y_k et c_k par l'itération (4).

Ce résultat montre que dans la méthode des multiplicateurs la suite $\{\|y_k - \bar{y}\|\}$ converge au moins linéairement si c_k est borné supérieurement et Q superlinéairement si c_k tend vers l'infini. (Pour les renseignements concernant la vitesse de convergence, voir par exemple [11] Orthega et Rheinboldt (1970)). Par contre, avec la méthode de pénalisation quadratique ($y_k = 0$), la vitesse de convergence est beaucoup moins favorable.

B. Au point de vue calcul, il est intéressant de ne faire la minimisation du lagrangien augmenté seulement qu'approximativement, en augmentant la précision de l'approximation à chaque étape.

C'est ce que nous considérerons également et nous obtiendrons des estimations similaires à (7) et (8) pour deux critères d'arrêts différents.

C. Nous utiliserons ces résultats pour construire une théorie de la dualité globale dans l'esprit de celle proposée par Rockafellar.

Nous montrerons que la fonctionnelle duale est

continuellement différentiable et que sa valeur, ainsi que son gradient peuvent être calculés par une minimisation sans contrainte du lagrangien augmenté (5).

D. Le problème (1) ne comporte que des contraintes d'égalité.

L'analyse cependant s'appliquera dans son entièreté aux contraintes d'inégalité puisque celles-ci peuvent être converties en contraintes d'égalité en utilisant des variables d'écart.

b. Les hypothèses pour lesquelles l'analyse est valable

Considérons le problème de programmation non linéaire:

$$\begin{aligned} & \text{minimiser } f(x) \\ & \text{sous contraintes : } h_i(x) = 0 \quad i=1, \dots, m \end{aligned} \quad (9)$$

où $f, h_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad i=1, \dots, m$

Soit \bar{x} une solution optimale du problème (9). On fait les hypothèses suivantes concernant f et h_i , sur une boule ouverte $B(\bar{x}, \varepsilon)$:

(A) : Le point \bar{x} , avec son vecteur \bar{y} des multiplicateurs de Lagrange, satisfait les conditions suffisantes de second ordre pour que \bar{x} soit un minimum local, c'est-à-dire :

(A1) : Les fonctions $f, h_i \quad i=1, \dots, m$ sont deux fois continuellement différentiables dans la boule $B(\bar{x}, \varepsilon)$.

(A2) : Les gradients $\nabla h_i(\bar{x})$, $i=1, \dots, m$ sont linéairement indépendants et il existe un vecteur unique des multiplicateurs de Lagrange $\bar{y} = (\bar{y}^1, \dots, \bar{y}^m)$ tel que :

$$\nabla f(\bar{x}) + \sum_{i=1}^m \bar{y}^i \nabla h_i(\bar{x}) = 0$$

(A3) : La matrice Hessienne du lagrangien $L_0(x, y)$ au point (\bar{x}, \bar{y}) :

$$\nabla^2 L_0(\bar{x}, \bar{y}) = \nabla^2 f(\bar{x}) + \sum_{i=1}^m \bar{y}^i \nabla^2 h_i(\bar{x})$$

est définie positive sur le plan tangent correspondant aux contraintes, c'est-à-dire :

$$w' \nabla^2 L_0(\bar{x}, \bar{y}) w > 0$$

pour tout $w \in \mathbb{R}^n$ tel que $w \neq 0$, $w' \nabla h_i(\bar{x}) = 0$, $i=1, \dots, m$.

(B) : Les matrices Hessiennes $\nabla^2 f$, $\nabla^2 h_i$ sont lipschitz continues à l'intérieur d'une boule ouverte $B(\bar{x}, \varepsilon)$, c'est-à-dire qu'il existe un $K > 0$ tel que :

pour tout $x, x' \in B(\bar{x}, \varepsilon)$

$$\|\nabla^2 f(x) - \nabla^2 f(x')\| \leq K \|x - x'\|$$

et

$$\|\nabla^2 h_i(x) - \nabla^2 h_i(x')\| \leq K \|x - x'\| \quad i=1, \dots, m$$

où $\|\cdot\|$ désigne la norme Euclidienne habituelle.

c. L'algorithme de la méthode de pénalisation généralisée

Soit S un sous-ensemble borné arbitraire de \mathbb{R}^m .
 Considérons un scalaire $c > 0$ et un vecteur $y \in S$ et
 le lagrangien augmenté :

$$L(x, y, c) = f(x) + y'h(x) + c/2 \|h(x)\|^2 \quad (10)$$

où $h'(x) = (h_1(x), \dots, h_m(x))$, (" ' " désigne la transposition.)

Considérons l'algorithme suivant :

Pas 1 : étant donné $c_k > 0$, $y_k \in S$,
 trouver un point x_k minimisant (peut - être approximativement) la fonction $L(\cdot, y_k, c_k)$ définie
 par (10).

Pas 2 : déterminer $c_{k+1} > 0$, $y_{k+1} \in S$ en fonction de
 x_k, y_k, c_k suivant une certaine procédure.
 Retourner au pas 1.

d. Interprétation de l'algorithme

Il est facile de vérifier que pour tout x appartenant à \mathbb{R}^n , nous avons :

$$L(x, y_k, c_k) \geq f(x) + \frac{c_k}{4} \|h(x)\|^2 - \frac{1}{c} \|y_k\|^2$$

En effet :

$$L(x, y_k, c_k) = f(x) + \sum_{i=1}^m y_k^i h_i(x) + \sum_{i=1}^m [h_i(x)]^2$$

$$\begin{aligned}
&= f(x) + \frac{c_k}{2} \left(\frac{2}{c_k} \sum_{i=1}^m y_k^i h_i(x) + \frac{1}{2} \sum_{i=1}^m [h_i(x)]^2 + \right. \\
&\quad \left. + \frac{2}{(c_k)^2} \sum_{i=1}^m (y_k^i)^2 \right) - \frac{1}{c_k} \sum_{i=1}^m (y_k^i)^2 + \sum_{i=1}^m \frac{c_k}{4} [h_i(x)]^2 \\
&= f(x) + \frac{c_k}{2} \sum_{i=1}^m \left(\frac{1}{\sqrt{2}} h_i(x) + \frac{\sqrt{2}}{c_k} y_k^i \right)^2 - \\
&\quad - \frac{1}{c_k} \sum_{i=1}^m (y_k^i)^2 + \sum_{i=1}^m \frac{c_k}{4} [h_i(x)]^2 \\
&\geq f(x) - \frac{1}{c_k} \sum_{i=1}^m (y_k^i)^2 + \frac{c_k}{4} \|h_i(x)\|^2
\end{aligned}$$

Ainsi, lorsque c_k tend vers l'infini, $L(x, y_k, c_k)$ tend vers l'infini, pour toutes les suites $\{y_k\}$ incluses dans S et tous les vecteurs non admissibles x .

On peut donc imaginer une méthode qui consisterait à minimiser $L(\cdot, y_k, c_k)$ pour une suite de scalaires $\{c_k\}$, c_k tendant vers l'infini et une suite de vecteurs $\{y_k\}$, y_k appartenant à S . On espère ainsi obtenir un point admissible (car s'il était non-admissible, à la limite $L(x, y_k, c_k)$ tendrait vers l'infini) qui minimise $f(x)$.

Une question très intéressante également, est de déterminer des méthodes avantageuses pour ajuster y_k , de façon à avoir une convergence plus rapide. Dans le chapitre I, on verra qu'il vaut mieux choisir le vecteur y_k aussi proche que possible du vecteur optimal des multiplicateurs de Lagrange \bar{y} . On montrera que si $x(y, c)$ minimise $L(\cdot, y, c)$ alors le vecteur

$$\tilde{y} = y + c h[x(y, c)] \quad (11)$$

est une approximation du multiplicateur de Lagrange \bar{y} dans le sens que

$$\lim_{c \rightarrow \infty} \tilde{y} = \bar{y}$$

Donc on peut particulariser le pas 2 de l'algorithme en employant pour ajuster les vecteurs y_k , après une minimisation donnant x_k , l'itération ci-dessus (11). C'est celle utilisée dans la méthode des multiplicateurs.

En plus on montrera que cette itération donne une vitesse de convergence plus rapide que celle de la méthode de pénalisation ordinaire et que pour qu'elle converge vers le multiplicateur \bar{y} , il n'est pas nécessaire que c_k tende vers l'infini ; ce qui est également un avantage car on sait que pour la méthode de pénalisation quadratique, on a l'instabilité numérique lorsque c_k tend vers l'infini (voir par exemple pour cela [8] Lootsma (1970)).

CHAPITRE I

LA CONVERGENCE DE L'ALGORITHMEA. Proposition 1

Il existe un scalaire $c_1^* > 0$ tel que pour tout $c > c_1^*$ et tout $y \in S$, le lagrangien augmenté

$$L(x, y, c) = f(x) + \sum_{i=1}^m y^i h_i(x) + \frac{c}{2} \sum_{i=1}^m [h_i(x)]^2 \quad (12)$$

a un point unique de minimum $x(y, c)$ par rapport à x dans une boule centrée en \bar{x} .

En outre, il existe un scalaire $M_1 > 0$, tel que

$$\|x(y, c) - \bar{x}\| \leq M_1 \left\| \frac{y - \bar{y}}{c} \right\| \quad (13)$$

et

$$\|\tilde{y}(y, c) - y\| \leq M_1 \left\| \frac{y - \bar{y}}{c} \right\| \quad (14)$$

pour tout $c > c_1^*$ et tout $y \in S$

et où le vecteur $\tilde{y}(y, c) \in \mathbb{R}^m$ est donné par

$$\tilde{y}(y, c) = y + c h[x(y, c)] \quad (15)$$

B. Interprétation de la proposition 1a) Résultats au point de vue de l'ajustement des y_k

1) méthode de pénalisation quadratique

($y_k = 0$ pour tout k et en supposant donc que $0 \in S$)

La proposition donne les estimations suivantes :

$$\|x(0, c_k) - \bar{x}\| \leq M_1 \frac{\|\bar{y}\|}{c_k}$$

$$\|\tilde{y}(0, c_k) - \bar{x}\| \leq M_1 \frac{\|\bar{y}\|}{c_k}$$

On a donc la convergence si c_k tend vers l'infini et les suites $x(0, c_k)$, $\tilde{y}(0, c_k)$ tendent respectivement vers \bar{x} et \bar{y} .

2) méthode des multiplicateurs

Le vecteur est ajusté par l'itération :

$$y_{k+1} = \tilde{y}(y_k, c_k) = y_k + c_k h[x(y_k, c_k)]$$

cette itération est celle de la proposition (voir (15)).

D'après celle - ci , la convergence est plus rapide puisqu'on obtient les estimations suivantes :

$$\|\tilde{y}(y_k, c_k) - \bar{y}\| \leq M_1 \frac{\|y_k - \bar{y}\|}{c_k}$$

$$\|x(y_k, c_k) - \bar{x}\| \leq M_1 \frac{\|y_k - \bar{y}\|}{c_k}$$

Remarque

Pour garantir que la suite y_k reste bornée, on effectuera l'ajustement uniquement si

$$y_k + c_k [h[x(y_k, c_k)]] \in S$$

Sinon

$$y_{k+1} = y_k$$

c'est - à - dire que y_k est inchangé.

b) Résultats au point de vue de l'ajustement des scalaires c_k

1) méthode de pénalisation quadratique

Comme on l'a déjà signalé plus haut , pour avoir la convergence , c_k doit tendre vers l'infini.

2) méthode des multiplicateurs

* si c_k tend vers l'infini

$$\lim_{k \rightarrow \infty} \frac{\|y_{k+1} - \bar{y}\|}{\|y_k - \bar{y}\|} = 0$$

c'est - à - dire que la suite $\{y_k\}$ converge vers \bar{y} Q-superlinéairement (voir Orthega et Rheinboldt (1970)).

* si c_k tend vers c , un scalaire fini (où c est assez grand pour assurer que $c > M_1$, $c > c_1^*$ et que $y_k + c h(x_k) \in$ à une boule ouverte centrée en \bar{y} et contenue dans S).

$$\limsup_{k \rightarrow \infty} \frac{\|y_{k+1} - \bar{y}\|}{\|y_k - \bar{y}\|} \leq \frac{M_1}{c}$$

c'est - à - dire que la suite $\{y_k\}$ converge vers \bar{y} au moins linéairement avec un coefficient de convergence inversément proportionnel à c .

c) Conclusion : les avantages de la méthode des multiplicateurs

1) Convergence globale

La méthode des multiplicateurs converge à partir d'un point de départ arbitraire à l'intérieur d'un ensemble S borné pourvu que c_k soit suffisamment grand.

2) c_k n'a pas besoin de croître jusqu'à l'infini : on peut ainsi éviter les problèmes d'instabilité numérique comme ^{c'est le cas} dans la méthode de pénalisation quadratique.

3) la vitesse de convergence est plus rapide que celle de la méthode de pénalisation quadratique ordinaire.

Exemple

$$\text{Si } c_k = s^k \quad s > 1$$

. pour la méthode de pénalisation , on a que :

$$\|x(0, c_k) - \bar{x}\| \leq M_1 \|\bar{y}\| s^{-k}$$

. pour la méthode des multiplicateurs , avec $y_0 = 0$:

$$\|x(y_k, c_k) - \bar{x}\| \leq M_1^{k+1} \|\bar{y}\| s^{-(1+2+\dots+k)}$$

Le rapport des deux membres de droite des inégalités ci-dessus est :

$$\frac{M_1 \|\bar{y}\| s^{-k}}{M_1^{k+1} \|\bar{y}\| s^{-(1+2+\dots+k)}} = \prod_{i=0}^{k-1} \frac{s^i}{M_1}$$

Lorsque k tend vers l'infini, ce rapport tend vers l'infini également, d'où l'avantage évident de la méthode des multiplicateurs.

d) Remarque

Pour éviter de créer des impressions fausses, il faut souligner que la propriété de convergence globale de la méthode des multiplicateurs dépend de la méthode employée pour la minimisation sans contrainte de $L(., y_k, c_k)$.

En effet, une méthode particulière employée pour la minimisation sans contrainte du lagrangien augmenté génère une suite de points $x(y_k, c_k)$ qui sont des minima locaux de $L(., y_k, c_k)$, proches de \bar{x} . Mais $L(., y_k, c_k)$ peut avoir d'autres points de minimum local vers lesquels la méthode de minimisation sans contrainte peut être attirée et, à moins qu'après un certain indice la méthode de minimisation sans contrainte reste dans le voisinage du même point de minimum local du problème (9), notre analyse de la convergence n'est pas valide et il n'y a pas de raison de croire que la méthode des multiplicateurs soit meilleure ou pire que la méthode de pénalisation quadratique.

Soulignons cependant qu'en pratique, on utilise souvent le dernier point x_k de la $k^{\text{ème}}$ minimisation comme point de départ de la $(k+1)^{\text{ème}}$ minimisation et que cela contribue beaucoup à produire des suites $\{x_k\}$ qui sont proches du même point de minimum local du problème (9).

C. Démonstration de la proposition 1

La démonstration se fait en deux étapes : d'abord nous démontrons la proposition sous l'hypothèse supplémentaire :

(C) : La matrice hessienne de la fonction lagrangienne ordinaire :

$$\nabla^2 L_0(\bar{x}, \bar{y}) = \nabla^2 f(\bar{x}) + \sum_{i=1}^m \bar{y}^i \nabla^2 h_i(\bar{x})$$

est une matrice définie positive (c'est-à-dire qu'on a la convexité locale stricte).

Après on étendra la démonstration au cas général.

Première partie

On suppose donc que l'hypothèse (C) a lieu ; pour tout $x \in B(\bar{x}, \varepsilon)$ et pour y fixé $\in S$, $c > 0$, considérons les variables auxiliaires :

$$\begin{aligned} p &= x - \bar{x} \\ q &= y + c h(x) - \bar{y} \end{aligned}$$

où $h(x)' = (h_1(x), \dots, h_m(x))$

Voici l'idée générale de la démonstration dans le cas où l'hypothèse (C) a lieu : pour $c > 0$ et $y \in S$ donnés, nous cherchons donc un point $x(y, c)$ qui est un minimum de $L(., y, c)$.

Pour ce point $x(y,c)$, on aura donc que

$$\nabla L(x(y,c), y, c) = 0$$

En remplaçant dans $\nabla L(x,y,c)$, $\nabla f(x)$ et $\nabla h_i(x)$, $i=1, \dots, m$ par leur développement arrêté au premier ordre, on obtiendra un système d'équations en p et q .

On démontrera ensuite en utilisant deux lemmes de Polyak [13], que ce système a une solution unique satisfaisant une certaine propriété de laquelle nous pourrions déduire les estimations (13) et (14).

Enfin, on montrera que cette solution est bien un minimum local du lagrangien augmenté en vérifiant que sa matrice Hessienne est définie positive.

Pour tout $x \in B(\bar{x}, \varepsilon)$ nous avons :

$$\nabla f(x) = \nabla f(\bar{x}) + \nabla^2 f(\bar{x})p + r_1(p) \quad (17)$$

$$\nabla h_i(x) = \nabla h_i(\bar{x}) + \nabla^2 h_i(\bar{x})p + r_2^i(p) \quad i=1, \dots, m \quad (18)$$

où r_1 et r_2^i sont des fonctions de p à valeurs dans \mathbb{R}^n vérifiant :

$$r_1(0) = r_2^i(0) = 0 \quad i=1, \dots, m$$

$$\nabla r_1(p) = \nabla^2 f(x) - \nabla^2 f(\bar{x})$$

$$\nabla r_2^i(p) = \nabla^2 h_i(x) - \nabla^2 h_i(\bar{x})$$

En effet $r_1(p) = \nabla f(x) - \nabla f(\bar{x}) - \nabla^2 f(\bar{x})p$ donc $r_1(0) = 0$
et $\nabla r_1(p) = \nabla^2 f(x) + 0 - \nabla^2 f(\bar{x})$

idem pour $\nabla r_2^i(p)$ $i=1, \dots, m$

Par l'hypothèse (B) (condition de Lipschitz), nous avons que pour tout p tel que $\|p\| < \xi$:

$$\|\nabla r_1(p)\| \leq K \|p\| \quad (20)$$

$$\|\nabla r_2^i(p)\| \leq K \|p\| \quad i=1, \dots, m \quad (21)$$

Considérons maintenant le lagrangien augmenté $L(x, y, c)$. En dérivant on a :

$$\begin{aligned} \nabla L(x, y, c) &= \nabla f(x) + \sum_{i=1}^m y^i \nabla h_i(x) + c \sum_{i=1}^m h_i(x) \nabla h_i(x) \\ &= \nabla f(x) + \sum_{i=1}^m (y^i + c h_i(x)) \nabla h_i(x) \end{aligned}$$

(16), (17) et (18) donnent :

$$\begin{aligned} \nabla L(x, y, c) &= \nabla f(\bar{x}) + \nabla^2 f(\bar{x})p + r_1(p) + \\ &+ \sum_{i=1}^m (q^i + \bar{y}^i) (\nabla h_i(\bar{x}) + \nabla^2 h_i(\bar{x})p + r_2^i(p)) \\ &= \nabla^2 f(\bar{x})p + \sum_{i=1}^m (q^i + \bar{y}^i) \nabla^2 h_i(\bar{x})p + \\ &+ \nabla f(\bar{x}) + r_1(p) + \sum_{i=1}^m (q^i + \bar{y}^i) [\nabla h_i(\bar{x}) + r_2^i(p)] \end{aligned}$$

soit :

$$\begin{aligned} \nabla L(x, y, c) &= \nabla^2 L_0(\bar{x}, \bar{y})p + \nabla h(\bar{x})q + r_1(p) + \\ &+ \sum_{i=1}^m (q^i + \bar{y}^i) r_2^i(p) + \sum_{i=1}^m q^i \nabla^2 h_i(\bar{x})p \end{aligned}$$

où $\nabla h(\bar{x})$ est la matrice $n \times m$ dont les colonnes sont les $\nabla h_i(\bar{x})$, $i=1, \dots, m$.

On peut évidemment mettre $L(x, y, c)$ sous la forme :

$$\nabla L(x, y, c) = \nabla^2 L_0(\bar{x}, \bar{y})p + \nabla h(\bar{x})q + r_3(p, q) \quad (22)$$

où $r_3(p, q) \in \mathbb{R}^n$ est le vecteur défini par :

$$r_3(p, q) = r_1(p) + \sum_{i=1}^m (q^i + \bar{y}^i) r_2^i(p) + \sum_{i=1}^m q^i \nabla^2 h_i(\bar{x})p \quad (23)$$

Nous avons également que (16) implique :

$$\frac{q + \bar{y} - y}{c} = h(x) = h(\bar{x}) + \nabla h(\bar{x})'p + r_4(p)$$

ou , ce qui revient au même :

$$\nabla h(\bar{x})'p - \frac{1}{c} q = \frac{1}{c} (\bar{y} - y) - r_4(p) \quad (24)$$

où la fonction $r_4 : B(0, \varepsilon) \rightarrow \mathbb{R}^m$ vérifie :

$$r_4(0) = 0 \quad (25)$$

$$\nabla r_4^i(p) = \nabla h_i(x) - \nabla h_i(\bar{x}) \quad i=1, \dots, m$$

Utilisons à nouveau l'hypothèse (B) - condition de Lipschitz - pour obtenir :

$$\begin{aligned} \|\nabla r_4^i(p)\| &= \|\nabla h_i(x) - \nabla h_i(\bar{x})\| \\ &= \|\nabla^2 h_i(x')p\| \quad \text{où } x' \text{ est un point intermédiaire entre } x \text{ et } \bar{x} \\ &\leq \|\nabla^2 h_i(x')\| \cdot \|p\| \\ &\leq \|\nabla^2 h_i(x') - \nabla^2 h_i(\bar{x}) + \nabla^2 h_i(\bar{x})\| \cdot \|p\| \\ &\leq (K \cdot \varepsilon + \|\nabla^2 h_i(\bar{x})\|) \cdot \|p\| \end{aligned} \quad (26)$$

Combinons maintenant (22) et (23) : en effet, pour qu'un point $x \in B(\bar{x}, \xi)$ satisfasse

$$\nabla L(x, y, c) = 0$$

il est équivalent que le point correspondant $s = (p', q)'$, donné par (16) résolve les équations suivantes :

$$\nabla^2 L_0(\bar{x}, \bar{y}) p + \nabla h(\bar{x}) q = -r_3(p, q) \quad (27)$$

$$\nabla h(\bar{x})' p - \frac{1}{c} q = \frac{1}{c} (\bar{y} - y) - r_4(p)$$

ou sous forme plus compacte :

$$As = t + r(s)$$

où on a utilisé les notations suivantes :

$$A = \begin{pmatrix} \nabla^2 L_0(\bar{x}, \bar{y}) & \nabla h(\bar{x}) \\ \nabla h(\bar{x})' & -\frac{I}{c} \end{pmatrix} \quad s = \begin{pmatrix} p \\ q \end{pmatrix}$$

où $I =$ matrice identité $m \times m$

$$t = \begin{pmatrix} 0 \\ \frac{\bar{y} - y}{c} \end{pmatrix} \quad r(s) = \begin{pmatrix} -r_3(p, q) \\ -r_4(p) \end{pmatrix} \quad (28)$$

Maintenant, voyons que pour $r(s)$, on a des propriétés analogues à celles obtenues pour $r_1(p)$, $r_2(p)$ et $r_4(p)$

1) (19), (23), (24) et (16) donnent :

$$r(0) = 0 \quad (29)$$

En effet

$$r_4(0) = 0$$

$$r_3(0,0) = r_1(0) + \sum_{i=1}^m (0 + \bar{y}^i) r_2^i(0) + \sum_{i=1}^m 0 \cdot \nabla^2 h_i(\bar{x})_0 = 0$$

2) En outre, pour un s correspondant à un $x \in B(\bar{x}, \varepsilon)$, (20), (21), (23), (26) et (28) donnent, après calculs :

$$\|\nabla r(s)\| \leq \lambda \|s\| \quad (30)$$

où $\lambda > 0$ est une constante ne dépendant que de ε .
En effet :

$$\begin{aligned} \|\nabla r(s)\|^2 &= \|\nabla r_3(p, q)\|^2 + \|\nabla r_4(p)\|^2 \\ &\leq \|\nabla r_3(p, q)\|^2 + (K\varepsilon + \|\nabla^2 h_i(\bar{x})\|)^2 \|p\|^2 \\ &\leq \|\nabla r_3(p, q)\|^2 + \beta^2 \|s\|^2 \end{aligned} \quad (31)$$

où $\beta^2 = (K\varepsilon + \|\nabla^2 h_i(\bar{x})\|)^2$ ne dépend que de ε .

$$\|s\|^2 = \|p\|^2 + \|q\|^2$$

Il reste à calculer $\nabla r_3(p, q)$ et à majorer la norme :

$$\begin{aligned} \nabla r_3(p, q) &= \left(\nabla r_1(p) + \sum_{i=1}^m (q^i + \bar{y}^i) \nabla r_2^i(p) + \sum_{i=1}^m q^i \nabla^2 h_i(\bar{x}) \right) \\ &\quad \left[(r_2^1(p), \dots, r_2^m(p)) + (\nabla^2 h_1(\bar{x})p, \dots, \nabla^2 h_m(\bar{x})p) \right]' \end{aligned} \quad \times$$

$$\begin{aligned} \|\nabla r_3(p, q)\|^2 &\leq \|\nabla r_1(p) + \sum_{i=1}^m (q^i + \bar{y}^i) \nabla r_2^i(p) + \sum_{i=1}^m q^i \nabla^2 h_i(\bar{x})\|^2 + \\ &\quad + \|r_2^1(p) + \nabla^2 h_1(\bar{x})p \dots r_2^m(p) + \nabla^2 h_m(\bar{x})p\|^2 \end{aligned}$$

Majorons d'abord le premier terme du membre de droite de l'inégalité ci-dessus :

$$\begin{aligned}
 & \|\nabla r_1(p) + \sum_{i=1}^m (q^i + \bar{y}^i) \nabla r_2^i(p) + \sum_{i=1}^m q^i \nabla^2 h_i(\bar{x})\| \\
 & \leq \|\nabla r_1(p)\| + \sum_{i=1}^m |q^i| \cdot \|\nabla r_2^i(p)\| + \sum_{i=1}^m |\bar{y}^i| \cdot \|\nabla r_2^i(p)\| + \\
 & \quad + \sum_{i=1}^m |q^i| \|\nabla^2 h_i(\bar{x})\| \\
 & \leq K \|p\| + \sum_{i=1}^m |q^i| \cdot K \|p\| + \sum_{i=1}^m |\bar{y}^i| K \|p\| + \\
 & \quad + \sum_{i=1}^m \|q\| \|\nabla^2 h_i(\bar{x})\| \\
 & \leq K \|p\| + C \|q\| \cdot K \|p\| + C \|\bar{y}\| K \|p\| + \|q\| \sum_{i=1}^m \|\nabla^2 h_i(\bar{x})\|
 \end{aligned}$$

où C est la constante intervenant dans l'inégalité exprimant l'équivalence des normes :

$$\begin{aligned}
 & |x_1| + |x_2| + \dots + |x_n| \leq C \|x\| \quad , \quad x \in \mathbb{R}^n \\
 & \leq K \|s\| + CK \|s\| + C \|\bar{y}\| K \|s\| + \|s\| \sum_{i=1}^m \|\nabla^2 h_i(\bar{x})\| \\
 & \leq (K + CK + C \|\bar{y}\| K + \sum_{i=1}^m \|\nabla^2 h_i(\bar{x})\|) \|s\| \\
 & \leq \gamma \|s\|
 \end{aligned}$$

$$\text{où } \gamma = K + CK + C \|\bar{y}\| K + \sum_{i=1}^m \|\nabla^2 h_i(\bar{x})\|$$

et γ ne dépend que de ε .

(32) donne alors :

$$\|\nabla r_3(p, q)\|^2 \leq \gamma^2 \|s\|^2 + \|(r_2^1(p) + \nabla^2 h_1(\bar{x})p \dots r_2^m(p) + \nabla^2 h_m(\bar{x})p)\|^2 \quad (33)$$

Majorons maintenant le second terme du membre de droite de l'inégalité ci-dessus :

$$\begin{aligned} & \|(r_2^1(p) + \nabla^2 h_1(\bar{x})p \dots r_2^m(p) + \nabla^2 h_m(\bar{x})p)\| \\ & \leq \sum_{i=1}^m \|r_2^i(p) + \nabla^2 h_i(\bar{x})p\| \end{aligned} \quad (34)$$

et on a que :

$$\begin{aligned} \|r_2^i(p) + \nabla^2 h_i(\bar{x})p\| &= \|\nabla h_i(\bar{x}) - \nabla h_i(x)\| \\ &\leq \beta \|s\| \end{aligned}$$

où β est défini plus haut.

(33) et (34) donnent :

$$\begin{aligned} \|\nabla r_3(p, q)\|^2 &\leq \gamma^2 \|s\|^2 + \sum_{i=1}^m \beta^2 \|s\|^2 \\ &\leq (\gamma^2 + m\beta^2) \|s\|^2 \end{aligned}$$

On obtient finalement :

$$\begin{aligned} \|\nabla r(s)\|^2 &\leq (\gamma^2 + m\beta^2) \|s\|^2 + \beta^2 \|s\|^2 \\ &\leq \alpha^2 \|s\|^2 \end{aligned}$$

où $\alpha = \sqrt{\gamma^2 + (m+1)\beta^2} > 0$ et ne dépend que de ε .

Maintenant , on va démontrer que (27) a une solution unique à l'intérieur du domaine de définition de s pour $y \in S$ et $c > c_1^*$, où c_1^* est une constante suffisamment grande.

Pour faire cela , on utilise les 2 lemmes suivants ; on trouvera leurs démonstrations dans Polyak [13] .

Lemme 1

La matrice A de (28) a un inverse pour tout $c > 0$. En outre , l'inverse est uniformément bornée , c'est - à - dire qu'il existe un scalaire $M_1 > 0$ tel que pour tout $c > 0$

$$\|A^{-1}\| \leq 2 M_1 \quad (35)$$

Lemme 2

L'équation (27) , $As = t + r(s)$ a une solution unique s^* dans la boule ouverte $B(0, 8M_1 \|t\|) \subset B(0, \varepsilon)$ pour tout $y \in S$ et tout c suffisamment grand pour garantir que

$$\|t\| \leq \min \left\{ \frac{1}{16 M_1 \alpha} , \frac{\varepsilon}{8 M_1} \right\}$$

où α et M_1 sont définis dans (30) et (35).

La solution s^* satisfait :

$$\|s^*\| \leq M_1 \|t\|$$

Ainsi d'après le lemme 2 , pourvu que c soit suffisamment grand , il existe un s^* unique satisfaisant l'équation

$$As = t + r(s) ,$$

ou d'une manière équivalente , d'après (27) il existe un point $x(y,c)$ unique satisfaisant

$$\nabla L(x(y,c), y, c) = 0$$

En outre

$$\|s^*\| \leq M_1 \|t\| \quad (36)$$

avec $t = \begin{pmatrix} 0 \\ \frac{y - \bar{y}}{c} \end{pmatrix}$ donc $\|t\| = \frac{\|y - \bar{y}\|}{c}$

$$\begin{aligned} \|s^*\|^2 &= \|p^*\|^2 + \|q^*\|^2 \\ &= \|x(y,c) - \bar{x}\|^2 + \|y + c h(x) - \bar{y}\|^2 \end{aligned}$$

On obtient donc , d'après (36) :

$$\|x(y,c) - \bar{x}\| \leq M_1 \frac{\|y - \bar{y}\|}{c}$$

$$\|y + c h(x) - \bar{y}\| \leq M_1 \frac{\|y - \bar{y}\|}{c}$$

Ce sont les inégalités qui étaient à prouver.

Par conséquent pour achever la preuve de la première partie de la proposition 1 , nous n'avons plus

qu'à montrer que pour c suffisamment grand le point $x(y,c)$ est un minimum local de $L(.,y,c)$.

Pour cela, il suffit de montrer que la matrice Hessienne du lagrangien augmenté en ce point $x(y,c)$ est définie positive pour tout $y \in S$ et tout c suffisamment grand.

Calculons d'abord $\nabla^2 L(x(y,c), y, c)$

$$\begin{aligned} \nabla^2 L(x(y,c), y, c) &= \nabla^2 f[x(y,c)] + \\ &+ \sum_{i=1}^m (y^i + c h_i[x(y,c)]) \nabla^2 h_i[x(y,c)] \\ &+ c \nabla h[x(y,c)] \cdot \nabla h[x(y,c)] \quad (37) \end{aligned}$$

Le troisième terme du membre de droite de l'égalité ci-dessus :

$$c \nabla h[x(y,c)] \cdot \nabla h[x(y,c)]$$

est une matrice semi-définie positive. Il suffit donc de montrer que :

$$\nabla^2 f[x(y,c)] + \sum_{i=1}^m (y^i + c h_i[x(y,c)]) \nabla^2 h_i[x(y,c)]$$

est définie positive.

Or, d'après l'hypothèse (C),

$$\nabla^2 L_0(\bar{x}, \bar{y}) = \nabla^2 f(\bar{x}) + \sum_{i=1}^m \bar{y}^i \nabla^2 h_i(\bar{x})$$

est définie positive et par continuité, il existe une boule ouverte $B(\bar{x}, \delta)$ et une boule ouverte $B(\bar{y}, \delta)$ telles que pour tout $x \in B(\bar{x}, \delta)$ et tout $y \in B(\bar{y}, \delta)$,

$$\nabla^2 f(x) + \sum_{i=1}^m y^i \nabla^2 h_i(x)$$

est définie positive.

Puisqu'on a déjà démontré que pour c suffisamment grand on a :

$$\|x(y,c) - \bar{x}\| \leq \frac{M_1}{c} \|y - \bar{y}\|$$

et

$$\|y + c h[x(y,c)] - \bar{y}\| \leq \frac{M_1}{c} \|y - \bar{y}\|$$

si c est suffisamment grand $x(y,c) \in B(\bar{x}, \delta)$ et $y + ch[x(y,c)] \in B(\bar{y}, \delta)$ et donc on a bien que

$$\nabla^2 f[x(y,c)] + \sum_{i=1}^m (y^i + c h_i[x(y,c)]) \nabla^2 h_i[x(y,c)]$$

est une matrice définie positive.

Ainsi, $\nabla^2 L[x(y,c), y, c]$ est la somme d'une matrice définie positive et d'une matrice semi-définie positive donc est une matrice définie positive et $x(y,c)$ est bien un minimum local de $L(., y, c)$.

Nous avons ainsi terminé la première partie de la démonstration, c'est - à - dire que nous avons prouvé la proposition 1 sous l'hypothèse supplémentaire (C).

Etendons maintenant la démonstration au cas général.

Deuxième partie

On va changer le problème (9) de programmation non linéaire général en un problème localement convexe équi-

valent pour lequel la condition (C) est satisfaite.

On obtient la convexité locale en ajoutant un terme de pénalisation assez grand à la fonction objective.

Il est évident que le problème (9) est équivalent pour tout $\mu \geq 0$ au problème suivant :

$$\begin{aligned} & \text{minimiser } f(x) + \frac{\mu}{2} \|h(x)\|^2 \\ & \text{sous les contraintes : } h_i(x) = 0 \quad i=1, \dots, m \end{aligned} \quad (38)$$

Le problème (38) a \bar{x} comme solution optimale et \bar{y} comme vecteur optimal des multiplicateurs de Lagrange. En effet :

$$\begin{aligned} L_\mu(x, y) &= f(x) + \frac{\mu}{2} \|h(x)\|^2 + y'h(x) \\ \text{et} \\ \nabla L_\mu(\bar{x}, \bar{y}) &= \nabla f(\bar{x}) + \mu \nabla h(\bar{x}) \cdot h(\bar{x}) + \nabla h(\bar{x}) \cdot \bar{y} \\ &= \nabla f(\bar{x}) + \nabla h(\bar{x}) \bar{y} = \nabla L_0(\bar{x}, \bar{y}) \end{aligned}$$

où $L_0(x, y)$ est le lagrangien ordinaire du problème (9).

Maintenant , considérons la matrice Hessienne par rapport à x du Lagrangien ordinaire du problème (38).

Nous avons :

$$\begin{aligned} \nabla^2 L_\mu(\bar{x}, \bar{y}) &= \nabla^2 f(\bar{x}) + \mu [\nabla h(\bar{x}) \nabla h(\bar{x})'] + \sum_{i=1}^m \bar{y}^i \nabla^2 h_i(\bar{x}) + \\ &+ \sum_{i=1}^m h_i(\bar{x}) \nabla^2 h_i(\bar{x}) \\ &= \nabla^2 L_0(\bar{x}, \bar{y}) + \mu [\nabla h(\bar{x}) \nabla h(\bar{x})'] \end{aligned} \quad (39)$$

En utilisant l'hypothèse (A3) , on obtient le lemme suivant :

Lemme 3

Il existe un scalaire $\mu^* \geq 0$ tel que pour tout $\mu > \mu^*$, la matrice $\nabla^2 L_\mu(\bar{x}, \bar{y})$ définie par (39) est définie positive.

Nous allons démontrer cela , c'est - à - dire nous allons démontrer que

$$w' \nabla^2 L_\mu(\bar{x}, \bar{y}) w > 0 \quad \text{pour tout } w \neq 0 \in \mathbb{R}^n$$

ou encore que

$$w' \nabla^2 L_0(\bar{x}, \bar{y}) w + \mu w' [\nabla h(\bar{x}) \nabla h(\bar{x})'] w > 0 \quad \text{pour tout } w \neq 0$$

et comme ceci est équivalent à démontrer que

$$\frac{w' \nabla^2 L_0(\bar{x}, \bar{y}) w}{\|w\|} + \mu \frac{w' [\nabla h(\bar{x}) \nabla h(\bar{x})'] w}{\|w\|} > 0 \quad \text{pour tout } w \neq 0 \in \mathbb{R}^n$$

il suffira de démontrer qu'il existe $\mu^* \geq 0$ tel que pour tout $\mu > \mu^*$

$$w' \nabla^2 L_0(\bar{x}, \bar{y}) w + \mu w' [\nabla h(\bar{x}) \nabla h(\bar{x})'] w > 0 \quad \text{pour tout } w \in \mathbb{R}^n \\ \text{tel que } \|w\| = 1$$

1) Supposons d'abord que w est tel que $\|w\| = 1$ et

$$w' \nabla h(\bar{x}) = 0$$

Alors on a que :

$$w' \nabla^2 L_\mu(\bar{x}, \bar{y}) w = w' \nabla^2 L_0(\bar{x}, \bar{y}) w$$

or d'après l'hypothèse (A3) , $\nabla^2 L_0(\bar{x}, \bar{y})$ est définie positive et donc la thèse est démontrée pour tout $\mu \geq 0$.

2) Par continuité de $\nabla^2 L_0(x, y)$, il existe η tel que pour tout $w \in V$ avec $\|w\| = 1$, où

$$V = \{w \in \mathbb{R}^n \mid |w' \nabla h(\bar{x})| < \eta\}$$

$$w' \nabla^2 L_0(\bar{x}, \bar{y}) w > 0$$

Ceci entraîne que

$$w' \nabla^2 L_\mu(\bar{x}, \bar{y}) w = w' \nabla^2 L_0(\bar{x}, \bar{y}) w + \mu w' [\nabla h(\bar{x}) \nabla h(\bar{x})'] w > 0$$

et la thèse est démontrée pour tout $\mu \geq 0$.

3) Maintenant dans le cas où $x \notin V$, posons :

$$\mu^* = \sup_{\substack{w \notin V \\ \|w\| = 1}} \left\{ - \frac{w' \nabla^2 L_0(\bar{x}, \bar{y}) w}{w' [\nabla h(\bar{x}) \nabla h(\bar{x})'] w} \right\}$$

Comme on prend le sup sur un ensemble compact , il n'est pas infini et on a trouvé un $\mu^* \geq 0$ vérifiant la thèse.

La conséquence immédiate de ce lemme est que le problème (38) satisfait la condition de convexité locale pour tout $\mu > \mu^*$. On peut alors appliquer le résultat de la proposition 1 qui a été prouvé sous la condition (C) ,

en remplaçant c par $c - \mu^*$ pour le problème :

$$\begin{aligned} \min \quad & f(x) + \frac{\mu^*}{2} \|h(x)\|^2 \\ & h(x) = 0 \end{aligned} \quad (40)$$

En effet considérons $L_{\mu^*}(x, y, c)$.

$$\begin{aligned} L_{\mu^*}(x, y, c) &= f(x) + \frac{\mu^*}{2} \|h(x)\|^2 + y'h(x) + \frac{c}{2} \|h(x)\|^2 \\ &= f(x) + y'h(x) + \frac{c + \mu^*}{2} \|h(x)\|^2 \end{aligned}$$

d'où

$$L_{\mu^*}(x, y, c - \mu^*) = L(x, y, c)$$

où $L(x, y, c)$ est le lagrangien augmenté du problème (9).

En appliquant la proposition 1, on obtient le résultat suivant :

Il existe $c^* > 0$ tel que pour tout $c - \mu^* > c^*$ et tout $y \in S$, le lagrangien augmenté du problème (9) $L(x, y, c)$ a un minimum sans contrainte unique $x(y, c)$ dans une boule ouverte centrée en \bar{x} .

En outre, il existe une constante $M > 0$ telle que

$$\|x(y, c) - \bar{x}\| \leq M \frac{\|y - \bar{y}\|}{c - \mu^*} \quad (41)$$

et

$$\|y + c h[x(y, c)] - \mu^* h[x(y, c)] - \bar{y}\| \leq M \frac{\|y - \bar{y}\|}{c - \mu^*} \quad (42)$$

L'inégalité (42) peut encore s'écrire :

$$\|y(y, c) - \bar{y} - \delta(y, c)\| \leq M \frac{\|y - \bar{y}\|}{c - \mu^*}$$

$$\text{où } \delta(y, c) = \mu^* h[x(y, c)]$$

$$\text{et } \tilde{y}(y, c) = y + c h[x(y, c)]$$

$$\|\delta(y, c)\| = \mu^* \|h[x(y, c)]\| = \mu^* \|h[x(y, c)] - h(\bar{x})\|$$

et comme

$$\|x(y, c) - \bar{x}\| \leq M \frac{\|y - \bar{y}\|}{c - \mu^*}$$

il existe une constante B telle que

$$\|\delta(y, c)\| \leq \mu^* B \frac{\|y - \bar{y}\|}{c - \mu^*}$$

On obtient l'estimation suivante :

$$\begin{aligned} \|\tilde{y}(y, c) - \bar{y}\| &= \|\tilde{y}(y, c) - \bar{y} - \delta(y, c) + \delta(y, c)\| \\ &\leq \|\tilde{y}(y, c) - \bar{y} - \delta(y, c)\| + \|\delta(y, c)\| \\ &\leq M \frac{\|y - \bar{y}\|}{c - \mu^*} + \mu^* B M \frac{\|y - \bar{y}\|}{c - \mu^*} \\ &\leq (M + \mu^* B M) \frac{\|y - \bar{y}\|}{c - \mu^*} \end{aligned} \quad (43)$$

Posons maintenant :

$$M_1 = (M + \mu^* B M) \frac{(c^* + \mu^*)}{c^*}$$

$$c_1^* = c^* + \mu^*$$

Pour tout $c > c_1^*$ on a :

$$\frac{M}{c - \mu^*} \leq \frac{M + \mu^* B M}{c - \mu^*} \leq \frac{M_1}{c}$$

La première inégalité est évidente. Vérifions la seconde .
En remplaçant M_1 par sa valeur , on obtient l'inégalité :

$$\frac{M + \mu^* B M}{c - \mu^*} \leq \frac{(M + \mu^* B M)(c^* + \mu^*)}{c c^*}$$

qui est équivalente à toutes les inégalités suivantes :

$$\frac{1}{c - \mu^*} \leq \frac{c^* + \mu^*}{c c^*}$$

$$c c^* \leq (c^* + \mu^*)(c - \mu^*)$$

$$c c^* \leq c^* c - c^* \mu^* + \mu^* c - \mu^{*2}$$

$$0 \leq -c^* + c - \mu^*$$

ce qui est bien vrai pour tout $c > c_1^*$

D'après (41) et (43) , pour tout $c > c_1^*$, on obtient donc les estimations suivantes :

$$\|x(y, c) - \bar{x}\| \leq M_1 \left\| \frac{y - \bar{y}}{c} \right\|$$

$$\|\tilde{y}(y, c) - \bar{y}\| \leq M_1 \left\| \frac{y - \bar{y}}{c} \right\|$$

Ce sont les estimations qu'on devait prouver.
La démonstration de la proposition est donc totalement terminée.

CHAPITRE II

LA CONVERGENCE DE L'ALGORITHME DANS LE CAS D'UNE
 MINIMISATION NON EXACTE DU LAGRANGIEN AUGMENTE

Nous allons donc maintenant analyser la méthode de pénalisation généralisée dans laquelle, étant donné c_k et y_k , le lagrangien augmenté $L(., y_k, c_k)$ n'est pas minimisé exactement mais où le processus de minimisation est considéré comme terminé lorsqu'un certain critère d'arrêt est vérifié.

Nous considérons deux critères d'arrêt différents :

- 1) la minimisation de $L(., y_k, c_k)$ est terminée au point x_k satisfaisant :

$$\| \nabla L(x_k, y_k, c_k) \| \leq \frac{\gamma_k}{c_k}$$

où $\{\gamma_k\}$ est une suite bornée avec $\gamma_k \geq 0$.

- 2) ou alors la minimisation est terminée au point x_k satisfaisant :

$$\| \nabla L(x_k, y_k, c_k) \| \leq \min \left\{ \frac{\gamma_k}{c_k}, \gamma_k', \|h(x_k)\| \right\}$$

où $\{\gamma_k\}$, $\{\gamma_k'\}$ sont bornées, avec $\gamma_k, \gamma_k' \geq 0$

Les résultats concernant la convergence de cette méthode sont rassemblés dans la proposition 2.

A. Proposition 2

Il existe un scalaire $c_2^* \geq 0$ tel que pour tout $c > c_2^*$, pour tout $y \in S$ et pour tout vecteur $a \in \mathbb{R}^n$ avec

$$\|a\| \leq \frac{\gamma_k}{c}$$

où $\{\gamma_k\}$ est une suite bornée de nombres positifs ou nuls, il existe un point unique $x_a(y, c)$ dans une boule ouverte centrée en \bar{x} , satisfaisant :

$$\nabla L(x_a(y, c), y, c) = a$$

En outre il existe un scalaire M_2 tel que :

$$\|x_a(y, c) - \bar{x}\| \leq M_2 \frac{(\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2}}{c}$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq M_2 \frac{(\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2}}{c}$$

où

$$\tilde{y}_a(y, c) = y + c h[x_a(y, c)]$$

Si en plus a et $x_a(y, c)$ satisfont

$$\|a\| \leq \gamma_k' \|h[x_a(y, c)]\|$$

pour tout $c > c_2^*$ et pour tout $y \in S$, il existe M_2 tel que

$$\|x_a(y, c) - \bar{x}\| \leq \frac{M_2}{c} \|y - \bar{y}\| (1 + 4(\gamma_k')^2)^{1/2}$$

$$\|y_a(y, c) - \bar{y}\| \leq \frac{M_2}{c} \|y - \bar{y}\| (1 + 4(\gamma_k')^2)^{1/2}$$

B. Interprétation de la Proposition 2

a) L'algorithme

A l'itération k , x_k est le point unique $x_a(y_k, c_k)$ correspondant à :

$$a = \nabla L(x_k, y_k, c_k)$$

c'est-à-dire que x_k est le point généré par la procédure de minimisation sans contrainte du lagrangien augmenté, celle-ci étant terminée lorsqu'est satisfait soit le critère d'arrêt 1 :

$$\nabla L(x_k, y_k, c_k) \leq \frac{\gamma_k}{c_k}$$

soit le critère d'arrêt 2 :

$$\nabla L(x_k, y_k, c_k) \leq \min \left\{ \frac{\gamma_k}{c_k}, \gamma_k \cdot \|h(x_k)\| \right\}$$

Ensuite c_k et y_k sont ajustés d'une certaine manière.

b) L'ajustement des multiplicateurs y_k

La proposition donne une procédure d'ajustement des multiplicateurs très avantageuse puisqu'elle accélère la convergence. Cette procédure est la même que celle proposée dans le cas d'une minimisation exacte et est donnée par l'itération :

$$y_{k+1} = y_k + c_k h(x_k)$$

Cet ajustement aura lieu uniquement si le point obtenu $\in S$.

c) La convergence

1) critère d'arrêt 2
.....

D'après la proposition, la suite $\{(x_k, y_k)\}$ générée par l'algorithme et cette itération, ainsi que le critère d'arrêt 2, converge vers (\bar{x}, \bar{y}) pourvu que c_k soit suffisamment grand et que \bar{y} appartienne à S .

De plus,

* quand c_k tend vers l'infini
la suite $\{y_k\}$ converge vers \bar{y} au moins Q -superlinéairement
* quand c_k reste fixé après un certain indice (en supposant qu'il est suffisamment grand),
la suite $\{y_k\}$ converge linéairement vers \bar{y} .

2) critère d'arrêt 1
.....

* quand c_k tend vers l'infini, on a également la convergence Q -superlinéaire.

* par contre quand c_k est gardé fixé après un certain indice, la convergence linéaire ne peut plus être garantie et de plus, à moins que $\{\gamma_k\}$ soit une suite convergente vers 0, pour atteindre la convergence globale, il faut nécessairement faire tendre c_k vers l'infini.

C. Démonstration de la Proposition 2

L'idée générale de la démonstration est la même

que celle de la proposition 1'. Nous démontrerons d'abord la proposition sous la condition (C). Ensuite, on étendra la démonstration au cas général.

Première partie

Supposons que l'hypothèse (C) est vérifiée. En reprenant le raisonnement de la démonstration de la proposition 1, et avec les mêmes notations, on a que :

$$\nabla^2 L(x, y, c) = \nabla^2 L_0(\bar{x}, \bar{y}) p + \nabla h(\bar{x}) q + r_3(p, q)$$

et

$$\nabla h(\bar{x})' p - \frac{1}{c} q = \frac{1}{c} (\bar{y} - y) - r_4(p)$$

pour tout $x \in B(\bar{x}, \xi)$, tout $y \in S$ et tout $c > 0$

Par conséquent, pour que le point $x_a(y, c)$, où $a \in \mathbb{R}^n$ est tel que

$$\|a\| \leq \frac{\gamma_k}{c} \quad \text{où } \{\gamma_k\} \text{ est une suite bornée et } \gamma_k > 0$$

satisfasse :

$$\nabla L(x_a(y, c), y, c) = a$$

il faut et il suffit que $x_a(y, c)$ soit solution du système suivant :

$$\begin{pmatrix} \nabla^2 L_0(\bar{x}, \bar{y}) & \nabla h(\bar{x}) \\ \nabla h(\bar{x})' & -\frac{I}{c} \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} a \\ \frac{\bar{y} - y}{c} \end{pmatrix} + \begin{pmatrix} -r_3(p, q) \\ -r_4(p) \end{pmatrix}$$

A

s

t_a

r(s)

Pour démontrer que ce système admet une solution et qu'elle est unique, nous allons à nouveau utiliser les lemmes 1 et 2 que nous avons cités dans le chapitre I. Le lemme 2 peut être appliqué puisque $\|a\| \leq \frac{\gamma_k}{c}$ et qu'on a supposé que $\{\gamma_k\}$ est une suite bornée.

On a donc qu'il existe $c_1^* \geq 0$ et $M_1 > 0$ tels que pour tout $y \in S$ et tout $c > c_1^*$, il existe un point $x_a(y, c)$ unique qui est solution de

$$\nabla L(x, y, c) = a$$

En outre le lemme 2 donne l'estimation suivante :

$$\|s^*\| \leq M_1 \|t_a\|$$

où s^* est la solution du système $As = t_a + r(s)$.
Ce qui donne :

$$(\|p^*\|^2 + \|q^*\|^2)^{1/2} \leq M_1 \left(\|a\|^2 + \frac{\|y - \bar{y}\|^2}{c^2} \right)^{1/2}$$

d'où

$$\|p\| \leq M_1 \left(\|a\|^2 + \frac{\|y - \bar{y}\|^2}{c^2} \right)^{1/2}$$

$$\|q\| \leq M_1 \left(\|a\|^2 + \frac{\|y - \bar{y}\|^2}{c^2} \right)^{1/2}$$

ou encore :

$$\|x_a(y, c) - \bar{x}\| \leq M_1 \left(\frac{\|y - \bar{y}\|^2}{c^2} + \|a\|^2 \right)^{1/2} \quad (51)$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq M_1 \left(\frac{\|y - \bar{y}\|^2}{c^2} + \|a\|^2 \right)^{1/2} \quad (52)$$

On obtient finalement :

$$\|x_a(y, c) - \bar{x}\| \leq \frac{M_1}{c} (\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2} \quad (53)$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq \frac{M_1}{c} (\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2} \quad (54)$$

Ce sont les relations qui étaient à prouver concernant le critère d'arrêt 1.

Maintenant, supposons en plus que a et x_a satisfont :

$$\|a\| \leq \gamma_k' \|h[x_a(y, c)]\|$$

De (51) et (52) on déduit :

$$\|x_a(y, c) - \bar{x}\| \leq M_1 \left(\frac{\|y - \bar{y}\|^2}{c^2} + (\gamma_k')^2 \|h[x_a(y, c)]\|^2 \right) \quad (55)$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq M_1 \left(\frac{\|y - \bar{y}\|^2}{c^2} + (\gamma_k')^2 \|h[x_a(y, c)]\|^2 \right) \quad (56)$$

D'autre part on a que :

$$\begin{aligned} \|c h[x_a(y, c)]\| &= \|c h[x_a(y, c)] + y - \bar{y} - (y - \bar{y})\| \\ &\leq \|c h[x_a(y, c)] + y - \bar{y}\| + \|y - \bar{y}\| \\ &\leq \|\tilde{y}_a(y, c) - \bar{y}\| + \|y - \bar{y}\| \\ &\leq M_1 \left(\frac{\|y - \bar{y}\|^2}{c^2} + (\gamma_k')^2 \|h[x_a(y, c)]\|^2 \right)^{1/2} + \|y - \bar{y}\| \\ &\leq M_1 \left(\frac{\|y - \bar{y}\|}{c} + \gamma_k' \|h[x_a(y, c)]\| \right) + \|y - \bar{y}\| \end{aligned}$$

$$\|c h[x_a(y, c)]\| \leq \left(\frac{M_1}{c} + 1\right) \frac{\|y - \bar{y}\|}{c} + M_1 \gamma_k' \|h[x_a(y, c)]\|$$

d'où :

$$\|h[x_a(y, c)]\| \leq \frac{c + M_1}{c(c - M_1 \gamma_k')} \|y - \bar{y}\| \quad (57)$$

Pour $c \geq (1 + 2\gamma_k')M_1$ on a $\frac{c + M_1}{c(c - M_1 \gamma_k')} \geq \frac{2}{c}$

(en effet si $c \geq (1 + 2\gamma_k')M_1$, on a $c - 2\gamma_k'M_1 \geq M_1$
donc $2c - 2\gamma_k'M_1 \geq c + M_1$ ou encore :

$$\frac{c + M_1}{c(c - M_1 \gamma_k')} \leq \frac{2}{c} \quad)$$

Combinant ceci avec (57) , si c est suffisamment grand on obtient :

$$\|h[x_a(y, c)]\| \leq 2 \frac{\|y - \bar{y}\|}{c}$$

Ainsi d'après (55) et (56) , pour c suffisamment grand on obtient les estimations suivantes :

$$\|x_a(y, c) - \bar{x}\| \leq M_1 \left(\frac{\|y - \bar{y}\|}{c^2} + (\gamma_k')^2 + 4 \frac{\|y - \bar{y}\|^2}{c^2} \right)^{1/2}$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq M_1 \left(\frac{\|y - \bar{y}\|}{c^2} + (\gamma_k')^2 + 4 \frac{\|y - \bar{y}\|^2}{c^2} \right)^{1/2}$$

Ce qui donne finalement :

$$\|x_a(y, c) - \bar{x}\| \leq \frac{M_1}{c} \left(1 + 4(\gamma_k')^2 \right)^{1/2} \|y - \bar{y}\|$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq \frac{M_1}{c} \left(1 + 4(\gamma_k')^2 \right)^{1/2} \|y - \bar{y}\|$$

Nous avons ainsi terminé la démonstration de la proposition 2 dans le cas où on a l'hypothèse (C). L'extention au cas général se fait de la même manière que pour la proposition 1.

Deuxième partie

Dans la proposition 1, on a obtenu que le problème (9) est équivalent au problème :

$$\begin{aligned} \min \quad & f(x) + \frac{\mu}{2} \|h(x)\|^2 \\ h_i(x) = & 0 \quad \quad \quad i=1, \dots, m \end{aligned}$$

pour tout $\mu > 0$

Pour ce problème, on a démontré qu'il existe $\mu^* > 0$ tel que pour tout $\mu > \mu^*$, la matrice Hessienne du lagrangien ordinaire pour ce problème est définie positive. Ce nouveau problème satisfait donc l'hypothèse (C) pour laquelle la proposition 2 a déjà été démontrée. On obtient alors les résultats suivants :

Il existe un scalaire $c_1^* > 0$ tel que pour tout $c > c_1^*$ et pour tout $y \in S$

$$\nabla L(x, y, c) = a \quad \text{où} \quad \|a\| \leq \frac{\gamma_k}{c}$$

a une solution unique $x_a(y, c)$ dans une boule ouverte centrée en \bar{x} .

En plus, on obtient qu'il existe une constante $M > 0$ telle que :

$$\|x_a(y, c) - \bar{x}\| \leq M \frac{(\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2}}{c - \mu^*}$$

$$\|y + c h[x_a(y, c)] - \mu^* h[x_a(y, c)] - \bar{y}\| \leq M \frac{(\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2}}{c - \mu^*}$$

Cette dernière inégalité, par un raisonnement analogue à celui fait pour la proposition 1, peut se transformer de la manière suivante :

$$\|\tilde{y}_a(y, c) - \bar{y} + \delta_a(y, c)\| \leq M \frac{(\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2}}{c - \mu^*}$$

où $\delta_a(y, c) = \mu^* h[x_a(y, c)]$

On a aussi qu'il existe une constante B telle que :

$$\begin{aligned} \|\delta_a(y, c)\| &\leq \mu^* B \|x_a(y, c) - \bar{x}\| \\ &\leq \mu^* B M \frac{(\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2}}{c - \mu^*} \end{aligned}$$

On obtient alors l'inégalité :

$$\begin{aligned} \|\tilde{y}_a(y, c) - \bar{y}\| &\leq M \frac{(\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2}}{c - \mu^*} + \|\delta_a(y, c)\| \\ &\leq \frac{(M + \mu^* B M)}{c - \mu^*} (\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2} \end{aligned}$$

Posons

$$c_2^* = c_1^* + \mu^* \geq 0$$

$$M_2 = (M + \mu^* B M) \frac{(c_1^* + \mu^*)}{c_1^*}$$

Pour tout $c > c_2^*$, on a que :

$$\frac{M}{c - \mu^*} \leq \frac{M + \mu^* B M}{c - \mu^*} \leq \frac{M_2}{c}$$

et on obtient finalement les inégalités qu'on voulait prouver :

$$\|x_a(y, c) - \bar{x}\| \leq \frac{M_2}{c} (\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2}$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq \frac{M_2}{c} (\|y - \bar{y}\|^2 + \gamma_k^2)^{1/2}$$

Maintenant si en plus $x_a(y, c)$ et a satisfont :

$$\|a\| \leq \gamma_k' \|h[x_a(y, c)]\|$$

alors on a les estimations :

$$\|x_a(y, c) - \bar{x}\| \leq M \left(\frac{\|y - \bar{y}\|^2}{(c - \mu^*)^2} + (\gamma_k')^2 \|h[x_a(y, c)]\|^2 \right)^{1/2}$$

$$\|y + (c - \mu^*) h[x_a(y, c)] - \bar{y}\| \leq M \left(\frac{\|y - \bar{y}\|^2}{(c - \mu^*)^2} + \right.$$

$$\left. + (\gamma_k')^2 \|h[x_a(y, c)]\|^2 \right)^{1/2}$$

En combinant les raisonnements employés dans la proposition 1 et dans la première partie de la proposition 2, on obtient :

$$(c - \mu^*) \|h[x_a(y, c)]\| \leq M \left(\frac{\|y - \bar{y}\|^2}{(c - \mu^*)^2} + (\gamma_k')^2 \|h[x_a(y, c)]\|^2 \right) + \|y - \bar{y}\|$$

$$\leq \left(\frac{M}{c - \mu^*} + 1 \right) \|y - \bar{y}\| + M \gamma_k' \|h[x_a(y, c)]\|$$

Ce qui donne :

$$\|h[x_a(y, c)]\| \leq \frac{M + c - \mu^*}{(c - \mu^*)(c - \mu^* - M \gamma_k')} \|y - \bar{y}\|$$

pour $c - \mu^* \geq (1 + 2 \gamma_k') M$ on a :

$$\frac{M + c - \mu^*}{(c - \mu^*)(c - \mu^* - M \gamma_k')} \leq \frac{2}{c - \mu^*}$$

donc

$$\|h[x_a(y, c)]\| \leq 2 \frac{\|y - \bar{y}\|}{c - \mu^*}$$

Pour tout c suffisamment grand, on obtient les estimations :

$$\|x_a(y, c) - \bar{x}\| \leq M \left(\frac{\|y - \bar{y}\|^2}{(c - \mu^*)^2} + (\gamma_k')^2 \frac{4}{(c - \mu^*)^2} \|y - \bar{y}\|^2 \right)^{\frac{1}{2}}$$

$$\|\tilde{y}_a(y, c) - \bar{y} - \mu^* h[x_a(y, c)]\|$$

$$\leq M \left(\frac{\|y - \bar{y}\|^2}{(c - \mu^*)^2} + (\gamma_k')^2 \frac{4}{(c - \mu^*)^2} \|y - \bar{y}\|^2 \right)^{\frac{1}{2}}$$

ou encore :

$$\|x_a(y, c) - \bar{x}\| \leq \frac{M}{c - \mu^*} \|y - \bar{y}\| (1 + 4(\gamma_k')^2)^{\frac{1}{2}}$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq \frac{M}{c - \mu^*} \|y - \bar{y}\| (1 + 4(\gamma_k')^2)^{\frac{1}{2}} + \mu^* \|h[x_a(y, c)]\|$$

comme il existe une constante B telle que

$$\|h[x_a(y, c)]\| \leq B \|x_a(y, c) - \bar{x}\|, \quad \text{on a :}$$

$$\|x_a(y, c) - \bar{x}\| \leq \frac{M}{c - \mu^*} \|y - \bar{y}\| (1 + 4(\gamma_k')^2)^{\frac{1}{2}}$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq \left(\frac{M}{c - \mu^*} + \mu^* B \frac{M}{c - \mu^*} \right) \|y - \bar{y}\| (1 + 4(\gamma_k')^2)^{\frac{1}{2}}$$

Posons

$$c_2^* = c_1^* + \mu^* \geq 0$$

$$M_2 = (M + \mu^* B M) \left(\frac{c_1^* + \mu^*}{c_1^*} \right)$$

alors pour tout $c > c_2^*$

$$\frac{M}{c - \mu^*} \leq \frac{M + \mu^* B M}{c - \mu^*} \leq \frac{M_2}{c}$$

Et on obtient les inégalités cherchées, c'est - à - dire pour tout $c > c_2^*$ et tout $y \in S$:

$$\|x_a(y, c) - \bar{x}\| \leq \frac{M_2}{c} \|y - \bar{y}\| (1 + 4(\gamma_k')^2)^{\frac{1}{2}}$$

$$\|\tilde{y}_a(y, c) - \bar{y}\| \leq \frac{M_2}{c} \|y - \bar{y}\| (1 + 4(\gamma_k')^2)^{\frac{1}{2}}$$

Ceci termine la démonstration de la proposition 2.

CHAPITRE III

LA METHODE DES MULTIPLICATEURS DANS UN CADRE DE
DUALITE GLOBALE

Introduction

Dans ce chapitre , on va utiliser les résultats des deux chapitres précédents pour construire un cadre de dualité pour le problème (9).

$$\begin{aligned} & \text{minimiser } f(x) \\ & \text{sous les contraintes } h_i(x) = 0 \quad i=1, \dots, m \end{aligned}$$

où $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $h_i: \mathbb{R}^n \rightarrow \mathbb{R}$ $i=1, \dots, m$

Un des résultats les plus importants est que la fonctionnelle duale est une fonction concave à valeurs réelles (finies) et partout définie. On démontrera aussi une autre propriété très importante : la différentiabilité de cette fonctionnelle.

On verra également que sa valeur et ses dérivées dans un ouvert arbitraire peuvent être calculées par une minimisation locale sans contrainte du lagrangien augmenté.

Enfin , on interprétera la méthode des multiplicateurs comme une itération du gradient au sens global.

A. Solution du problème non linéaire avec perturbation des contraintes

Pour un vecteur $u \in \mathbb{R}^m$, considérons le problème

de minimisation suivant :

$$\begin{aligned} & \text{minimiser } f(x) && (61) \\ & \text{sous les contraintes } h_i(x) = u_i \quad i=1, \dots, m \end{aligned}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $h_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad i=1, \dots, m$

Pour ce problème , on a le résultat suivant :

Lemme 4

Sous l'hypothèse (A) , il existe des scalaires β et δ tels que pour tout u tel que $\|u\| < \beta$, le problème (61) a une solution unique $x(u)$ dans la boule ouverte $B(\bar{x}, \delta)$, avec un multiplicateur de Lagrange $y(u)$ satisfaisant :

$$\|y(u) - \bar{y}\| < \delta$$

De plus , les fonctions $x(u)$ et $y(u)$ sont continuellement différentiables dans la boule ouverte $B(0, \beta)$ et satisfont :

$$\begin{aligned} x(0) &= \bar{x} \\ y(0) &= \bar{y} \end{aligned}$$

Ce lemme exprime que si le problème initial (9) a une solution vérifiant les conditions suffisantes de second ordre , le problème avec une petite perturbation sur les contraintes , a aussi une solution

unique qui a une propriété importante pour la suite de notre analyse , à savoir la différentiabilité continue de cette solution par rapport au paramètre mesurant la perturbation.

Ce lemme est une application immédiate du théorème des fonctions implicites dont on trouvera l'énoncé dans l'appendice A (voir par exemple Dieudonné [4]).

Pour appliquer ce théorème , et par conséquent démontrer le lemme , il suffit de vérifier que les conditions du théorème sont vérifiées , en faisant les correspondances suivantes :

- $u \in \mathbb{R}^m$ joue le rôle du x du théorème
- $(x,y) \in \mathbb{R}^n \times \mathbb{R}^m$ joue le rôle du y du théorème
- les $n+m$ fonctions suivantes , à valeurs réelles :

$$(x,y) \rightarrow \nabla f(x) + \sum_{i=1}^m y^i \nabla h_i(x)$$

$$(x,y) \rightarrow h_i(x) - u_i \quad i=1, \dots, m$$

jouent le rôle des g_i du théorème.

Elles sont continues et continument différentiables.

Vérifions maintenant les hypothèses :

- i) $u = 0$ joue le rôle du x_0 du théorème
- (\bar{x}, \bar{y}) joue le rôle du y_0 du théorème et

$$\nabla f(\bar{x}) + \sum_{i=1}^m \bar{y}^i \nabla h_i(\bar{x}) = 0$$

$$h_i(\bar{x}) = 0 \quad i=1, \dots, m$$

- ii) Les dérivées du système de fonctions ci-dessus par rapport à (x,y) existent et

sont continues.

iii) Le Jacobien du système est :

$$J = \begin{vmatrix} \nabla^L L_0(\bar{x}, \bar{y}) & \nabla h(\bar{x}) \\ \nabla h(\bar{x}) & 0 \end{vmatrix}$$

$J \neq 0$ puisque par l'hypothèse (A3) ,
 $\nabla^L L_0(\bar{x}, \bar{y})$ est définie positive sur le
 plan tangent correspondant aux contraintes
 et que les gradients des contraintes au
 point \bar{x} sont linéairements indépendants.

Toutes les hypothèses sont donc vérifiées et on
 peut appliquer le théorème des fonctions implicites
 pour obtenir immédiatement le lemme 4.

B. La fonctionnelle duale et ses propriétés

Définissons d'abord la fonctionnelle primale :

$$p : B(0, \beta) \rightarrow \mathbb{R}$$

$$\begin{aligned} p(u) &= \min_{\substack{f(x) = f[x(u)] \\ h(x) = u \\ x \in B(\bar{x}, \delta)}} f(x) &= f[x(u)] & (62) \end{aligned}$$

Comme par le théorème des fonctions implicites ,
 on a , pour tout $u \in B(0, \beta)$:

$$\nabla f[x(u)] + \nabla h[x(u)] \cdot y(u) = 0 \quad (a)$$

$$h[x(u)] = u \quad (b)$$

en transposant l'égalité (a) ci-dessus et en la multipliant par la matrice $n \times m$ dont les lignes sont les $\nabla_u x^i(u)$ $i=1, \dots, n$. On obtient alors :

$$\nabla p(u) + y(u)' \nabla_u h[x(u)] = 0$$

où $\nabla_u h[x(u)]$ est la matrice $m \times m$ dont les lignes sont les $\nabla_u h_i[x(u)]$ $i=1, \dots, m$. Or par l'égalité (b), cette matrice est la matrice identité $m \times m$.

On obtient donc que pour tout $u \in B(0, \rho)$:

$$\nabla p(u) = -y(u) \quad (63)$$

Puisque $y(u)$ est continuellement différentiable dans la boule $B(0, \rho)$, $p(u)$ est deux fois continuellement différentiable dans la boule $B(0, \rho)$. Sans perte de généralité, nous pouvons supposer que la matrice Hessienne de $p(u)$ est uniformément bornée dans cette boule (en effet, puisque la matrice Hessienne de $p(u)$ est continue, elle est uniformément bornée sur toute boule fermée contenue dans $B(0, \rho)$: il suffit alors de considérer une boule ouverte contenue dans une de ces boules fermées.)

Maintenant, pour $c \geq 0$ considérons la fonction :

$$p_c(u) = p(u) + \frac{c}{2} \|u\|^2$$

Pour c suffisamment grand, la matrice Hessienne de p_c est définie positive et donc p_c est strictement convexe sur $B(0, \rho)$. Pour un tel c , on définit la fonctionnelle duale :

$$d_c : \mathbb{R}^m \rightarrow \mathbb{R}$$

$$d_c(y) = \inf_{u \in B(0, \rho)} \left\{ p(u) + \frac{c}{2} \|u\|^2 + y'u \right\} = \inf_{u \in B(0, \rho)} \left\{ p_c(u) + y'u \right\} \quad (64)$$

La fonction d_c de (64) a les propriétés suivantes :

Proposition 3

Sous l'hypothèse (A) , pour chaque c pour lequel la matrice Hessienne de p_c est définie positive sur $B(0, \beta)$, nous avons les propriétés suivantes :

(a) La fonction d_c est une fonction concave, partout continuellement différentiable , à valeurs réelles. De plus , elle est deux fois continuellement différentiable sur l'ensemble ouvert

$$A = \{y \mid y = -\nabla p_c(u) , u \in B(0, \beta)\}$$

(b) Pour $y \in A$, l'infimum de (64) est atteint en un point unique $u_y \in B(0, \beta)$ et on a :

$$\nabla d_c(y) = u_y$$

$$\nabla^2 d_c(y) = -[\nabla^2 p_c(u_y)]^{-1}$$

(c) La fonctionnelle d_c a un point unique de maximum : le multiplicateur de Lagrange.

Démonstration

Dans la démonstration qui va suivre , on utilisera des notions d'analyse convexe (voir Rockafellar [16]) mais on trouvera cependant les références d'une manière précise. Pour une explication plus détaillée de l'u-

utilisation des théorèmes d'analyse convexe dans la démonstration de la différentiabilité continue de la fonctionnelle duale, voir appendice B.

(a) Considérons la fonction $g(u)$ qui prend la valeur :

$$g(u) = \begin{cases} p_c(u) & \text{sur } B(0, \beta) \\ + \infty & \text{ailleurs} \end{cases}$$

Cette fonction est strictement convexe.

Considérons aussi la fermeture de cette fonction :

$$(cl\ g)(u) = \sup_{(y, \mu^*) \in D} (y'u - \mu^*)$$

$$\text{où } D = \{ (y, \mu^*) \mid y \in \mathbb{R}^m, y'u - \mu^* \leq g(u); u \in \mathbb{R}^m \}$$

La fermeture est strictement convexe sur $B(0, \beta)$ (th.7.4) et le domaine effectif de cette fonction (l'ensemble des points où $(cl\ g)(u)$ est fini) est la boule fermée $\bar{B}(0, \beta)$, qui est un ensemble compact.

Considérons maintenant la fonction conjuguée de $cl\ g$:

$$(cl\ g)^*(y) = \sup_u (y'u - (cl\ g)(u))$$

cette dernière fonction est :

1) à valeurs réelles finies; en effet :

$$(cl\ g)^*(y) = \sup_u (y'u - (cl\ g)(u))$$

$$= \sup_{u \in \bar{B}(0, \beta)} (y'u - (cl\ g)(u))$$

or $y'u - (\text{cl } g)(u)$ est une fonction continue sur la boule compacte $\bar{B}(0, \beta)$.

2) convexe et continuellement différentiable (voir th.26.3 et cor.25.5.1)

De plus , on a que :

$$\begin{aligned} (\text{cl } g)(y) &= \sup_{u \in B(0, \beta)} (y'u - p_c(u)) = - \inf_{u \in B(0, \beta)} (p_c(u) - y'u) \\ &= - d_c(-y) \end{aligned}$$

On obtient donc que d_c est partout finie et continuellement différentiable ; puisque $(\text{cl } g)(y)$ est convexe , d_c est concave.

En appliquant le cor. 23.5.1 , on obtient

$$\begin{aligned} \nabla d_c [-\nabla p_c(u)] &= u \quad \text{pour tout } u \in B(0, \beta) \\ \nabla p_c [\nabla d_c(y)] &= -y \quad \text{pour tout } y \in A \end{aligned} \tag{66}$$

$$\begin{aligned} \text{où } A &= \{ y \mid y = -\nabla p_c(u) , u \in B(0, \beta) \} \\ &= \{ y \mid \nabla d_c(y) = u , u \in B(0, \beta) \} \end{aligned}$$

Puisque ∇d_c est continue , A est l'image réciproque de la boule ouverte $B(0, \beta)$ par une fonction continue : A est donc ouvert.

Soient maintenant \tilde{y} un point de A
et $\tilde{u} = \nabla d_c(\tilde{y})$

On a que : $\tilde{u} \in B(0, \beta)$

$$\nabla p_c(\tilde{u}) = -\tilde{y}$$

Nous allons appliquer le théorème des fonctions implicites à cette équation (voir énoncé appendice A)
On fait les correspondances suivantes :

théorème

notre cas particulier

x, x_0

y, \tilde{y}

y, y_0

u, \tilde{u}

ξ_i

$\nabla p_c(u) + y$

Toutes les hypothèses sont satisfaites :

1)

$$p_c(\tilde{u}) = -\tilde{y}$$

2) $\nabla^2 p_c(u)$ existe et est une fonction continue de y et de u

3)

$$|\nabla^2 p_c(\tilde{u})| \neq 0$$

puisque p_c est définie positive

4) $p_c(u)$ est continuellement différentiable dans $B(0, \beta)$

Il suit qu'il existe une boule ouverte $B(\tilde{y}, \lambda) \subset A$ et une fonction continuellement différentiable :

$$u(\cdot) : B(\tilde{y}, \lambda) \rightarrow B(0, \beta)$$

telle que :

$$u(\tilde{y}) = \tilde{u}$$

$$\nabla p_c[u(y)] = -y \quad \text{pour tout } y \in B(\tilde{y}, \lambda)$$

(66) donne alors :

$$\nabla d_c(y) = \nabla d_c \left[-\nabla p_c[u(y)] \right] = u(y) \quad \text{pour tout } y \in B(\tilde{y}, \lambda)$$

or $u(y)$ est continuellement différentiable sur $B(\tilde{y}, \lambda)$
donc $\nabla d_c(y)$ également.

Ainsi d_c est deux fois continuellement différentiable en \tilde{y} . Puisque \tilde{y} est un point arbitraire de A , on a que d_c est deux fois continuellement différentiable sur A : ce qu'on devait démontrer.

(b) 1) montrons d'abord que pour chaque $y \in A$,
l'infimum dans

$$\inf_{u \in B(0, \beta)} \left\{ p(u) + \frac{c}{2} \|u\|^2 + y'u \right\}$$

est atteint en un point unique $u_y \in B(0, \beta)$
en effet :

pour chaque $y \in A$, il existe $u_y \in B(0, \beta)$ tel que

$$y = -\nabla p_c(u_y)$$

d'où $\nabla (p_c(u) + y'u)(u_y) = \nabla p_c(u_y) + y = 0$

et on sait que

$$p_c(u) = p(u) + \frac{c}{2} \|u\|^2$$

est continue sur $B(0, \beta)$ et strictement convexe et donc $p_c(u) + u'y$ est également continue sur $B(0, \beta)$ et strictement convexe : u_y est donc un minimum unique.

Pour chaque $y \in A$, on obtient donc que :

$$\nabla d_c(y) = u_y$$

$$\nabla p_c(u_y) = -y$$

2) dérivons l'égalité :

$$\nabla d_c [-\nabla p_c(u)] = u$$

par rapport à u . Cela donne :

$$\nabla^2 d_c [-\nabla p_c(u)] \cdot (-1) \nabla^2 p_c(u) = 1 \quad \text{pour tout } u \in B(0, \rho)$$

On obtient finalement que pour chaque $y \in A$

$$\nabla^2 d_c(y) = -(\nabla^2 p_c(u_y))^{-1}$$

Et comme $\nabla^2 p_c(u)$ est définie positive pour tout $u \in B(0, \rho)$
 $\nabla^2 d_c(y)$ est définie négative pour tout $y \in A$.

$$(c) \nabla p_c(0) = \nabla p(0) + \frac{c}{2} 0 = \nabla p(0) = -y(0) = -\bar{y}$$

donc $u_{\bar{y}} = 0$ et on a que :

$$\nabla d_c(\bar{y}) = 0$$

et \bar{y} est le point de maximum unique de d_c puisque celle-ci est strictement concave.

C. Calcul de la valeur de la fonctionnelle duale ,
de ses dérivées partielles ainsi que de sa
matrice Hessienne

Maintenant , nous allons montrer que la valeur et les dérivées de la fonctionnelle duale peuvent être obtenues par une minimisation du lagrangien augmenté $L(.,y,c)$ pourvu que c soit suffisamment grand. Ceci nous est donné par la proposition suivante :

Proposition 4

Soit S un sous-ensemble ouvert borné de \mathbb{R}^m . Supposons que les hypothèses (A) et (B) aient lieu. Alors il existe un scalaire $c^* \geq 0$ tel que pour tout $y \in S$ et pour tout $c > c^*$, la fonctionnelle duale satisfait :

$$(a) \quad \nabla d_c(y) = h[x(y,c)]$$

où $x(y,c)$ est le point qui minimise localement $L(.,y,c)$.

$$(b) \quad d_c(y) = f[x(y,c)] + y'h[x(y,c)] + \frac{c}{2} \|h[x(y,c)]\|^2$$

$$= \min_x L(x,y,c)$$

(c) d_c est deux fois continuellement différentiable sur S et $\nabla^2 d_c(y)$ est donné par la formule :

$$\nabla^2 d_c(y) = -\nabla h[x(y,c)] \left\{ \nabla^2 L[x(y,c),y,c] \right\}^{-1} \nabla h[x(y,c)]$$

Démonstration

(a) Soit S un sous - ensemble ouvert borné de \mathbb{R}^m .
Alors pour tout $y \in S$, et pour tout c suffisamment grand
on a , d'après la proposition 1 :

$$\|x(y, c) - \bar{x}\| \leq M_1 \frac{\|y - \bar{y}\|}{c} < \delta$$

$$\|\tilde{y}(y, c) - \bar{y}\| \leq M_1 \frac{\|y - \bar{y}\|}{c} < \delta$$

où $\tilde{y}(y, c) = y + c h[x(y, c)]$

$$\tilde{u} = h[x(y, c)] \quad \|\tilde{u}\| < \rho$$

En outre , nous avons :

$$\nabla f[x(y, c)] + \sum_{i=1}^m \tilde{y}^i(y, c) \nabla h_i[x(y, c)] = 0$$

D'après le lemme 4 , $x(y, c)$ est le point de minimum
unique pour le problème :

$$\begin{aligned} \min & f(x) \\ h(x) &= \tilde{u} \\ x &\in B(\bar{x}, \delta) \end{aligned}$$

(62) et (63) donnent :

$$p(\tilde{u}) = f[x(y, c)]$$

$$\nabla p(\tilde{u}) = -\tilde{y}(y, c) = -y - c \tilde{u}$$

et donc $\nabla p_c(\tilde{u}) + y = \nabla p(\tilde{u}) + c \tilde{u} + y = 0$

Par conséquent $y \in A$ et \tilde{u} atteint l'infimum dans le membre de droite de

$$d_c(y) = \inf_{u \in B(0, \rho)} (p_c(u) + u'y)$$

et par la partie (b) de la proposition 3 :

$$d_c(y) = \tilde{u} = h[x(y, c)]$$

$$\begin{aligned} \text{(b)} \quad d_c(y) &= p(\tilde{u}) + y'\tilde{u} + \frac{c}{2}\|\tilde{u}\|^2 \\ &= f[x(y, c)] + y' h[x(y, c)] + \frac{c}{2} \|h[x(y, c)]\|^2 \\ &= \min_x L(x, y, c) \end{aligned}$$

où la minimisation est locale au sens de la proposition 1.

(c) Maintenant , il ne reste plus qu'à calculer

$$D_c(y) = \nabla^2 d_c(y)$$

Dérivons par rapport à y l'égalité : $\nabla d_c(y) = h[x(y, c)]$

$$\nabla^2 d_c(y) = \nabla h[x(y, c)] \cdot \nabla_y x(y, c)$$

où $\nabla h[x(y, c)]$ est la matrice $n \times m$ dont les colonnes sont les gradients $\nabla h_i[x(y, c)]$ $i=1, \dots, m$ et $\nabla_y x(y, c)$ la matrice $n \times m$ dont les lignes sont les gradients $\nabla_y x^i(y, c)$ par rapport à y $i=1, \dots, n$.

On a que :

$$\nabla L[x(y, c), y, c] = 0$$

c' est - à - dire :

$$\nabla f[x(y,c)] + \nabla h[x(y,c)] y + c h[x(y,c)] \nabla h[x(y,c)] = 0$$

Dérivons cette égalité par rapport à y :

$$\begin{aligned} \nabla^2 f[x(y,c)] \nabla_y x(y,c) + \nabla h[x(y,c)] + \sum_{i=1}^m y^i \nabla^2 h_i[x(y,c)] \nabla_y x(y,c) \\ + c \nabla h[x(y,c)] \nabla h[x(y,c)] \nabla_y x(y,c) + \\ + \sum_{i=1}^m c h_i[x(y,c)] \nabla^2 h_i[x(y,c)] \nabla_y x(y,c) = 0 \end{aligned}$$

ce qui donne :

$$\begin{aligned} \left[\nabla^2 f[x(y,c)] + \sum_{i=1}^m (y^i + c h_i[x(y,c)]) \nabla^2 h_i[x(y,c)] \right. \\ \left. + c \nabla h[x(y,c)] \nabla h[x(y,c)] \right] \nabla_y x(y,c) + \nabla h[x(y,c)] = 0 \end{aligned}$$

ou encore :

$$\nabla^2 L[x(y,c), y, c] \nabla_y x(y,c) + \nabla h[x(y,c)] = 0$$

On en déduit :

$$\nabla_y x(y,c) = - \left[\nabla^2 L[x(y,c), y, c] \right]^{-1} \nabla h[x(y,c)]$$

Remplaçons cette valeur de $\nabla_y x(y,c)$ dans $\nabla^2 d_c(y)$.

On obtient :

$$\nabla^2 d_c(y) = - \nabla h[x(y,c)] \cdot \left[\nabla^2 L[x(y,c), y, c] \right]^{-1} \nabla h[x(y,c)]$$

D. Interprétation de la méthode des multiplicateurs dans le cadre de la dualité

Puisque par la proposition précédente ,

$$d_c(y) = h[x(y, c)]$$

on peut écrire l'itération de la méthode des multiplicateurs , pour c suffisamment grand :

$$y_{k+1} = y_k + c_k \nabla d_c(y_k)$$

Cette itération peut alors être interprétée comme une itération du gradient à pas fixé pour maximiser la fonctionnelle duale d_c .

Ainsi on peut obtenir des résultats sur la vitesse de convergence en utilisant les résultats connus pour la méthode du gradient.

En particulier, lors de la mise en pratique de cette méthode , il faudra veiller à ce que c soit suffisamment grand pour assurer la convergence, mais d'autre part, il ne devra pas être trop grand pour ne pas obtenir un problème dual mal conditionné car les valeurs propres de D_c dépendent étroitement de d_c .

Cette interprétation primale-duale suggère aussi d'autres possibilités pour ajuster les multiplicateurs. L'une d'elles réside dans le fait que si les dérivées secondes sont calculées durant un cycle de minimisation sans contrainte, alors on obtient la matrice Hessienne $D_c(y) = \nabla^2 d_c(y)$ en plus du gradient ∇d_c . Il est alors possible d'exécuter une itération de Newton pour maximiser d_c au lieu de l'itération du gradient comme on le fait dans la méthode des multiplicateurs.

CHAPITRE IV

TRAITEMENT DES CONTRAINTES D'INEGALITE

Le traitement des contraintes d'inégalité peut se faire d'une manière très simple en introduisant des variables d'écart :

le problème :

$$\begin{aligned} \min f(x) \\ g_j(x) \leq 0 \quad j=1, \dots, r \end{aligned} \quad (I)$$

$$\text{où } f, g_j : \mathbb{R}^n \rightarrow \mathbb{R} \quad j=1, \dots, r$$

est équivalent au problème suivant, qui ne comporte que des contraintes d'égalité :

$$\begin{aligned} \min f(x) \\ g_j(x) + z_j^2 = 0 \quad j=1, \dots, r \end{aligned} \quad (II)$$

où z_1, \dots, z_r sont des variables additionnelles.

A. Validité de l'analyse pour les problèmes avec contraintes d'inégalité

Supposons que (\bar{x}, \bar{y}) est une paire de solution et de vecteur des multiplicateurs de Lagrange associé pour le problème (I) satisfaisant les conditions suffisantes de second ordre suivantes :

(A') : Les fonctions f, g_j $j=1, \dots, r$ sont deux fois continuellement différentiables dans la boule ouverte $B(\bar{x}, \varepsilon)$.

Les gradients des contraintes $\nabla g_j(\bar{x})$, $j \in J(\bar{x})$ où

$$J(\bar{x}) = \{ j \mid g_j(\bar{x}) = 0 \}$$

sont linéairement indépendants.

Nous avons que :

$$\nabla f(\bar{x}) + \sum_{j=1}^r \bar{y}^j \nabla g_j(\bar{x}) = 0$$

$$\bar{y}^j \geq 0$$

$$\bar{y}^j > 0 \Leftrightarrow j \in J(\bar{x})$$

En outre ,

$$w' \left[\nabla^2 f(\bar{x}) + \sum_{j=1}^r \bar{y}^j \nabla^2 g_j(\bar{x}) \right] w > 0$$

pour tout $w \neq 0$ tel que

$$w' \nabla g_j(\bar{x}) = 0 \text{ pour tout } j \in J(\bar{x})$$

On voit que $(\bar{x}', |g_1(\bar{x})|^{1/2}, \dots, |g_r(\bar{x})|^{1/2})' \in \mathbb{R}^{n+r}$ est solution optimale du problème (II).

Le point $((\bar{x}', |g_1(\bar{x})|^{1/2}, \dots, |g_r(\bar{x})|^{1/2}), \bar{y}') \triangleq ((\bar{x}', \bar{z}'), \bar{y}') \triangleq (\tilde{\bar{x}}', \bar{y}')$ $\in \mathbb{R}^{n+r} \times \mathbb{R}^r$ satisfait l'hypothèse (A) de l'introduction. En effet :

(A1) : les fonctions f , f_i pour $i=1, \dots, r$ où

$$f_i : \mathbb{R}^{n+r} \rightarrow \mathbb{R} \text{ et } f_i(x, z) = g_i(x) + z_i^2$$

sont deux fois continuellement différentiables dans la boule $B(\bar{x}, \varepsilon)$ par l'hypothèse (A').

(A2) : les gradients :

$$\nabla f_j(\bar{x}) = \begin{pmatrix} \nabla g_j(\bar{x}) \\ 0 \\ \vdots \\ 2|g_j(\bar{x})|^{1/2} \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} \in \mathbb{R}^n \\ \\ \\ \in \mathbb{R}^r \\ \\ \end{matrix}$$

pour $j = 1, \dots, r$,

sont linéairement indépendants puisque les $\nabla g_j(\bar{x})$ le sont.

Il existe un vecteur des multiplicateurs de Lagrange \bar{y} unique tel que

$$\begin{pmatrix} \nabla f(\bar{x}) \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{pmatrix} + \sum_{j=1}^r \bar{y}^j \begin{pmatrix} \nabla g_j(\bar{x}) \\ 0 \\ \vdots \\ 2|g_j(\bar{x})|^{1/2} \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$$

ou encore

$$\begin{pmatrix} \nabla f(\bar{x}) + \sum_{i=1}^r \bar{y}^i \nabla g_i(\bar{x}) \\ 2\bar{y}^1 |g_1(\bar{x})|^{1/2} \\ \vdots \\ 2\bar{y}^r |g_r(\bar{x})|^{1/2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$\bar{y}' = (\bar{y}^1, \dots, \bar{y}^r)$ existe : c'est celui du problème (I).

(A3) La matrice Hessienne du lagrangien ordinaire :

$$\tilde{L}_0(\tilde{x}, y) = f(x) + \sum_{j=1}^r y^j (g_j(x) + z_j^2)$$

est définie positive sur le plan tangent correspondant aux contraintes .

En effet :

$$\begin{aligned} \nabla^2 \tilde{L}_0(\tilde{x}, \bar{y}) &= \begin{pmatrix} \nabla^2 f(\bar{x}) & \circ \\ \circ & \circ \end{pmatrix} + \sum_{j=1}^r \bar{y}^j \begin{pmatrix} \nabla^2 g_j(\bar{x}) & & \circ \\ & 2 & \cdot \\ & & \cdot \\ & & & 2 \end{pmatrix} \\ &= \begin{pmatrix} f(\bar{x}) + \sum_{j=1}^r \bar{y}^j g_j(\bar{x}) & & & \circ \\ & \circ & & \\ & & 2 \bar{y}^1 & \\ & & & \cdot \\ & & & & 2 \bar{y}^r \end{pmatrix} \end{aligned}$$

Nous devons donc montrer que

$$\begin{aligned} \tilde{w} [\nabla^2 \tilde{L}_0(\tilde{x}, \bar{y})] \tilde{w}' > 0 \quad \text{pour tout } \tilde{w} \in \mathbb{R}^{n+r} \text{ tel que} \\ \tilde{w}' \neq 0 \\ \tilde{w}' \nabla f_j(\bar{x}) = 0 \quad j=1, \dots, r \end{aligned}$$

c'est - à - dire montrons que

$$\tilde{w} [\nabla^2 \tilde{L}_0(\tilde{x}, \bar{y})] \tilde{w}' = w [\nabla^2 L_0(\bar{x}, \bar{y})] w + \sum_{j=1}^r 2 \bar{y}^j (w_{n+j}^{\tilde{w}})^2 > 0$$

pour tout $\tilde{w} \in \mathbb{R}^{n+r}$, $\tilde{w} \neq 0$, $\tilde{w}' = (w, w_{n+1}^{\tilde{w}}, \dots, w_{n+r}^{\tilde{w}})$
tel que :

$$w' \nabla g_j(\bar{x}) + w_{n+j}^{\tilde{w}} \cdot 2 |g_j(\bar{x})|^{1/2} = 0 \quad j=1, \dots, r$$

Ce qui revient à montrer que :

$$w' \nabla^2 L_0(\bar{x}, \bar{y}) w > 0 \quad \text{pour tout } \tilde{w} \neq 0, w \in \mathbb{R}^{n+r}$$

$$\tilde{w} = (w, \tilde{w}_{n+1}, \dots, \tilde{w}_{n+r}), w \in \mathbb{R}^n \quad \text{tel que}$$

$$\begin{cases} w' \nabla g_j(\bar{x}) = 0 & j \in J(\bar{x}) & (a) \\ w' \nabla g_j(\bar{x}) + \tilde{w}_{n+j}^2 |g_j(\bar{x})|^{\frac{1}{2}} = 0 & j \notin J(\bar{x}) & (b) \end{cases}$$

Ceci est vrai d'après (A') pour les $\tilde{w} \in \mathbb{R}^{n+r}$ qui vérifient (a) donc à fortiori pour ceux qui vérifient (a) et (b).

Par conséquent, nous pouvons appliquer à ce nouveau problème toute la théorie que nous avons construite pour le problème à contraintes d'égalité à condition que l'hypothèse (B) de Lipschitz soit également vérifiée.

Ainsi on peut utiliser la méthode des multiplicateurs pour résoudre le problème (II) au lieu du problème (I).

B. Calcul du lagrangien augmenté, de l'itération des multiplicateurs et des critères d'arrêt.

Remarquons que les variables d'écart ne doivent pas figurer explicitement dans les calculs. En effet, on peut faire la minimisation du lagrangien augmenté $L(x, z, y, c)$ d'abord par rapport à z_1, \dots, z_r .

$$L(x, z, y, c) = f(x) + \sum_{i=1}^r y^i (g_i(x) + z_i^2) + \sum_{i=1}^r (g_i(x) + z_i^2)^2$$

$$\begin{aligned}\hat{L}(x, y, c) &= \min_z L(x, z, y, c) \\ &= f(x) + \sum_{i=1}^r y^i g_i(x) + \\ &\quad + \sum_{i=1}^r \min_z \left\{ y^i z_i^2 + \frac{c}{2} (g_i(x) + z_i^2)^2 \right\}\end{aligned}$$

Cherchons le $\min_z \left\{ y^i z_i^2 + \frac{c}{2} (g_i(x) + z_i^2)^2 \right\}$:

* dérivons le terme entre accolades par rapport à z_i et annulons cette dérivée :

$$2 y^i z_i + c(g_i(x) + z_i^2)2 z_i = 0$$

on obtient 2 racines :

$$1) z_i = 0$$

$$2) y^i + c g_i(x) + c z_i^2 = 0$$

d'où

$$z_i^2 = -\frac{y^i}{c} - g_i(x)$$

* la dérivée seconde doit être strictement positive pour que z_i soit minimum. Cette dérivée seconde est égale à :

$$\begin{aligned}& y^i + c g_i(x) + 2 c z_i^2 + c z_i^2 \\ &= y^i + c g_i(x) + 3 z_i^2\end{aligned}$$

$$\begin{aligned}1) z_i^2 = 0 \text{ est solution si } & y^i + c g_i(x) > 0 \\ & \text{c'est-à-dire si } -\frac{y^i}{c} - g_i(x) < 0\end{aligned}$$

2) $z_i^2 = -\frac{y^i}{c} - g_i(x)$ est solution si

$$\begin{aligned} & y^i + c g_i(x) + 3c \left(-\frac{y^i}{c} - g_i(x)\right) \\ &= y^i + c g_i(x) - 3y^i - 3c g_i(x) \\ &= -2 \left(y^i + c g_i(x)\right) > 0 \end{aligned}$$

c'est-à-dire si :

$$-\frac{y^i}{c} - g_i(x) > 0$$

En résumé la solution est :

$$z_i^2 = \max \left\{ 0, -\frac{y^i}{c} - g_i(x) \right\}$$

Calcul du lagrangien augmenté

$$\begin{aligned} \hat{L}(x, y, c) &= f(x) + \sum_{i=1}^r y^i \left(g_i(x) + \max \left\{ 0, -\frac{y^i}{c} - g_i(x) \right\} \right) \\ &\quad + \frac{c}{2} \left[g_i(x) + \max \left\{ 0, -\frac{y^i}{c} - g_i(x) \right\} \right]^2 \end{aligned}$$

1) si $-\frac{y^i}{c} - g_i(x) \leq 0$

le terme correspondant à l'indice i dans la somme est :

$$y^i g_i(x) + \frac{c}{2} g_i(x)^2$$

2) si $-\frac{y^i}{c} - g_i(x) > 0$

le terme correspondant à l'indice i dans la somme est :

$$\begin{aligned}
& y^i g_i(x) - \frac{(y^i)^2}{c} - y^i g_i(x) + \frac{c}{2} \left(-\frac{y^i}{c} - g_i(x) \right)^2 + \\
& + \frac{c}{2} [g_i(x)]^2 + 2 \frac{c}{2} g_i(x) \left(-\frac{y^i}{c} - g_i(x) \right)^2 \\
& = - \frac{(y^i)^2}{c} + \frac{c}{2} [g_i(x)]^2 + \frac{c}{2} \frac{(y^i)^2}{c^2} + \frac{c}{2} [g_i(x)]^2 \\
& + 2 \frac{c}{2} \frac{y^i}{c} g_i(x) - c g_i(x) \frac{y^i}{c} - c [g_i(x)]^2 \\
& = - \frac{(y^i)^2}{c}
\end{aligned}$$

Récapitulons : le terme correspondant à l'indice i dans la somme est égal à :

$$1) \text{ si } y^i + c g_i(x) \geq 0 \quad : \quad \left(\frac{y^i + c g_i(x)}{2c} \right)^2 - \frac{(y^i)^2}{2c}$$

$$2) \text{ si } y^i + c g_i(x) < 0 \quad : \quad 0 - \frac{(y^i)^2}{2c}$$

On obtient donc que :

$$\hat{L}(x, y, c) = f(x) + \frac{1}{2c} \sum_{i=1}^r \left([\max \{0, y^i + c g_i(x)\}]^2 - (y^i)^2 \right)$$

Maintenant, on peut minimiser $\hat{L}(x, y, c)$ par rapport à x et on obtient un vecteur $x(y, c)$.

Calcul de l'itération des multiplicateurs

$$\begin{aligned}
y_{k+1}^i &= y_k^i + c \left(g_i[x(y, c)] + z_i [x(y, c), y, c]^2 \right) \\
& \qquad \qquad \qquad i=1, \dots, r \\
&= y_k^i + c \left(g_i[x(y, c)] + \max \left\{ 0, -\frac{y_k^i}{c} - g_i[x(y, c)] \right\} \right)
\end{aligned}$$

$$1) \text{ si } -\frac{y_k^i}{c} - g_i[x(y,c)] \leq 0 \text{ (donc si } y_k^i + c g_i[x(y,c)] \geq 0 \text{)}$$

$$y_{k+1}^i = y_k^i + c g_i[x(y,c)]$$

$$2) \text{ si } -\frac{y_k^i}{c} - g_i[x(y,c)] > 0 \text{ (donc si } y_k^i + c g_i[x(y,c)] < 0 \text{)}$$

$$y_{k+1}^i = y_k^i + c g_i[x(y,c)] - c \frac{y_k^i}{c} - c g_i[x(y,c)] = 0$$

En résumé :

$$y_{k+1}^i = \max \{ 0, y_k^i + c g_i[x(y,c)] \}$$

Calcul du critère d'arrêt

$$\| \nabla \hat{L}(x_k, y_k, c_k) \| \leq \min \left\{ \frac{\gamma_k}{c_k}, \gamma_k, \sqrt{\sum_{i=1}^r (g_i(x_k) + z_i^2)^2} \right\}$$

$$\leq \min \left\{ \frac{\gamma_k}{c_k}, \gamma_k, \left[\sum_{i=1}^r \left(g_i(x_k) + \max \left\{ 0, \frac{-y_k^i}{c_k} - g_i(x_k) \right\} \right)^2 \right]^{1/2} \right\}$$

$$1) \text{ si } -\frac{y_k^i}{c_k} - g_i(x_k) \leq 0 \text{ donc si } -\frac{y_k^i}{c_k} \leq g_i(x_k)$$

$$g_i(x_k) + z_i^2 = g_i(x_k)$$

$$2) \text{ si } -\frac{y_k^i}{c_k} - g_i(x_k) > 0 \text{ donc si } g_i(x_k) > -\frac{y_k^i}{c_k}$$

$$g_i(x_k) + z_i^2 = g_i(x_k) - \frac{y_k^i}{c_k} - g_i(x_k) = -\frac{y_k^i}{c_k}$$

On peut rassembler ceci en écrivant :

$$g_i(x_k) + z_i^2 = \max \left\{ g_i(x_k), -\frac{y_k^i}{c_k} \right\}$$

D'où on obtient le critère d'arrêt :

$$\| \nabla \hat{L}(x_k, y_k, c_k) \| \leq \min \left\{ \frac{\gamma_k}{c_k}, \gamma_k \left(\sum_{i=1}^r [\max \{ g_i(x_k), -\frac{y_k^i}{c_k} \}]^2 \right)^{1/2} \right\}$$

Et ceci termine termine l'analyse de la méthode des multiplicateurs.

DEUXIEME PARTIE :

APPLICATION A DES PROBLEMES NON DIFFERENTIABLES :

UNE METHODE D'APPROXIMATION

Introduction

a. Philosophie de la méthode d'approximation par un problème différentiable

Dans cette partie, nous allons appliquer l'analyse de la méthode des multiplicateurs à des problèmes particuliers de minimisation où la fonction objective n'est pas différentiable.

Nous considérons les problèmes de la forme :

$$\begin{aligned} & \text{minimiser } g(x) \\ & x \in Q \subset \mathbb{R}^n \end{aligned} \quad (1)$$

où g est non différentiable mais seulement à cause de la présence dans son expression de termes de la forme :

$$\gamma[f_i(x)] = \max \{ 0, f_i(x) \} \quad i \in I \quad (2)$$

$$f_i : \mathbb{R}^n \rightarrow \mathbb{R}$$

On aimerait pouvoir appliquer les méthodes habituelles de minimisation mais pour cela, il faut que le problème soit différentiable. Une méthode consisterait donc à approximer chaque expression de la forme $\gamma[f_i(x)]$ par une fonction régulière et résoudre le problème différentiable obtenu par des méthodes bien connues. En faisant cela, on obtiendra une solution approchant la solution du problème et en rendant l'approximation de plus en plus précise, on espère qu'elle convergera vers la solution exacte.

b. Une forme d'approximation donnée par la méthode des multiplicateurs

En effet, considérons l'expression $\Upsilon[f(x)]$.
En utilisant une variable auxiliaire z , on peut écrire :

$$\begin{aligned} \Upsilon[f(x)] &= \min z \\ f(x) &\leq z \\ z &\geq 0 \end{aligned} \quad (3)$$

Nous pouvons appliquer à ce problème la méthode des multiplicateurs vue dans la première partie. Ainsi, utilisant le lagrangien augmenté, on obtient un problème de minimisation approximant $\Upsilon[f(x)]$:

$$\min_{z \geq 0} \left\{ z + \frac{1}{2c} \left(\left[\max \{0, y + c [f(x) - z]\} \right]^2 - y^2 \right) \right\} \quad (4)$$

avec $c > 0$ et $0 \leq y \leq 1$ (nous justifierons plus loin pourquoi $y \leq 1$)

Minimisons maintenant cette expression par rapport à z .

- 1) Supposons que pour x, y, c fixés $y + c f(x) \geq 0$
Considérons la fonction :

$$\mathbb{R} \rightarrow \mathbb{R}$$

$$z \mapsto z + \frac{1}{2c} \left(\left[y + c [f(x) - z] \right]^2 - y^2 \right) \quad (5)$$

et cherchons son minimum.

Annulant la dérivée , on obtient :

$$1 + \frac{1}{2c} 2(y + c [f(x) - z]) \cdot c \cdot (-1) = 0$$

$$y + c (f(x) - z) = 1$$

$$z = f(x) - \left(\frac{1 - y}{c} \right)$$

cette valeur de z est donc celle qui minimise la fonction définie par(5). On a deux cas :

$$(a) \quad \underline{f(x) - \left(\frac{1 - y}{c} \right) \leq 0}$$

dans ce cas $z = 0$ sera la solution du problème (4) et on obtient que :

$$\max \{ 0, y + c [f(x) - z] \} = y + c f(x)$$

ce qui donne en remplaçant z par sa valeur dans(4) :

$$\begin{aligned} \min \{ z + \frac{1}{2c} \left(\left[\max \{ 0, y + c [f(x) - z] \} \right]^2 - y^2 \right) \} \\ &= \frac{1}{2c} ([y + c f(x)]^2 - y^2) \\ &= \frac{1}{2c} (c^2 f(x)^2 + 2 y c f(x)) \\ &= \frac{c}{2} [f(x)]^2 + y f(x) \end{aligned}$$

$$(b) \quad \underline{f(x) - \left(\frac{1 - y}{c} \right) \geq 0}$$

alors $z = f(x) - \left(\frac{1 - y}{c} \right)$ est solution et

$$\begin{aligned} & \max \{ 0, y + c [f(x) - z] \} \\ & = \max \left\{ 0, y + c \left[f(x) - f(x) + \frac{(1-y)}{c} \right] \right\} \\ & = \max \{ 0, 1 \} = 1 \end{aligned}$$

et on obtient pour (4) :

$$\begin{aligned} & f(x) - \frac{(1-y)}{c} + \frac{1}{2c} (1-y^2) \\ & = f(x) - \frac{2(1-y) + 1-y^2}{2c} \\ & = f(x) - \frac{(1-y)^2}{2c} \end{aligned}$$

2) Supposons maintenant que $y + c f(x) \leq 0$

Alors pour tout $z \geq 0$

$$y + c [f(x) - z] \leq 0$$

et

$$\max \{ 0, y + c [f(x) - z] \} = 0$$

le problème (4) se réduit donc à :

$$\min_{z \geq 0} \left\{ z - \frac{y^2}{2c} \right\}$$

Le minimum par rapport à z est $z = 0$ et donc on obtient comme valeur pour (4) :

$$- \frac{y^2}{2c}$$

Si on rassemble tous ces résultats, le lagrangien augmenté du problème (3) nous donne une approximation de $\Upsilon[f(x)]$ qui s'exprime de la manière suivante :

$$\tilde{\Upsilon}[f(x), y, c] = \begin{cases} f(x) - \frac{(1-y)^2}{2c} & \text{si } f(x) \geq \frac{1-y}{c} \\ y f(x) + \frac{c}{2} [f(x)]^2 & \text{si } -\frac{y}{c} \leq f(x) \leq \frac{1-y}{c} \\ -\frac{y^2}{2c} & \text{si } f(x) \leq -\frac{y}{c} \end{cases} \quad (6)$$

Remarquons que si f est différentiable, alors $\tilde{\Upsilon}[f(x), y, c]$ est également différentiable. Son gradient est donné par :

$$\nabla \tilde{\Upsilon}[f(x), y, c] = \begin{cases} \nabla f(x) & \text{si } \frac{(1-y)}{c} \leq f(x) \\ [y + c f(x)] \nabla f(x) & \text{si } -\frac{y}{c} \leq f(x) \leq \frac{1-y}{c} \\ 0 & \text{si } f(x) \leq -\frac{y}{c} \end{cases} \quad (7)$$

c. Mesure de l'approximation de $\Upsilon(t)$ par $\tilde{\Upsilon}(t, y, c)$

Nous allons montrer que pour tout t , $c > 0$, $0 \leq y \leq 1$

$$\tilde{\Upsilon}(t, y, c) \leq \Upsilon(t) \leq \tilde{\Upsilon}(t, y, c) + \frac{1}{2c} \max \{ y^2, (1-y)^2 \} \quad (8)$$

1) Montrons que $\tilde{\Upsilon}(t, y, c) \leq \Upsilon(t)$ pour tout t .

Soit t appartenant à \mathbb{R} , arbitraire.

$$1^{\text{er}} \text{ cas : } 0 \leq \frac{1-y}{c} \leq t$$

$$\tilde{\gamma}(t, y, c) = t - \frac{(1-y)^2}{2c} \leq t = \gamma(t)$$

2^{ème} cas : $0 \leq t \leq \frac{(1-y)}{c}$

$$\tilde{\gamma}(t, y, c) = ty + \frac{1}{2} c t^2 = t \left(y + \frac{c}{2} t \right) \leq t = \gamma(t)$$

car $y + \frac{c}{2} t \leq y + ct \leq 1$

3^{ème} cas : $-\frac{y}{c} \leq t \leq 0$

$$\tilde{\gamma}(t, y, c) = ty + \frac{1}{2} ct^2 = t \underbrace{\left(y + \frac{c}{2} t \right)}_{\geq 0} \leq 0 = \gamma(t)$$

car $y + \frac{c}{2} t = \underbrace{y + ct}_{\geq 0} - \frac{ct}{2} \geq 0$

4^{ème} cas : $t \leq -\frac{y}{c}$

$$\tilde{\gamma}(t, y, c) = -\frac{y^2}{2c} \leq 0 = \gamma(t)$$

2) Montrons que

$$\gamma(t) \leq \tilde{\gamma}(t, y, c) + \frac{1}{2c} \max \{ y^2, (1-y)^2 \} \text{ pour tout } t$$

Soit t un réel arbitraire

1^{er} cas : $t \geq \frac{(1-y)}{c}$

$$\gamma(t) = t$$

$$\tilde{\gamma}(t, y, c) = t - \frac{(1-y)^2}{2c}$$

$$\Upsilon(t) = \tilde{\Upsilon}(t, y, c) + \frac{(1-y)^2}{2c}$$

2^{ème} cas : $0 \leq t \leq \frac{(1-y)}{c}$

$$\Upsilon(t) = t$$

$$\tilde{\Upsilon}(t, y, c) = y t + \frac{c t^2}{2}$$

montrons que $\Upsilon(t) = t \leq \tilde{\Upsilon}(t, y, c) + \frac{(1-y)^2}{2c}$

$$\tilde{\Upsilon}(t, y, c) + \frac{(1-y)^2}{2c} - t = y t + \frac{c t^2}{2} + \frac{(1-y)^2}{2c} - t$$

$$= \frac{1}{2c} (1 + y^2 + c^2 t^2 - 2y - 2c t + 2y c t)$$

$$= \frac{1}{2c} (1 - y - c t)^2 \geq 0$$

3^{ème} cas : $-\frac{y}{c} \leq t \leq 0$

$$\Upsilon(t) = 0$$

$$\tilde{\Upsilon}(t, y, c) = y t + \frac{c t^2}{2}$$

montrons que $0 = \Upsilon(t) \leq \tilde{\Upsilon}(t, y, c) + \frac{1}{2c} y^2$

$$\tilde{\Upsilon}(t, y, c) + \frac{1}{2c} y^2 = y t + \frac{c t^2}{2} + \frac{1}{2c} y^2$$

$$= \frac{1}{2c} (2c y t + c^2 t^2 + y^2)$$

$$= \frac{1}{2c} (c t + y)^2 \geq 0 = \Upsilon(t)$$

4^{ème} cas : $t \leq -\frac{y}{c}$

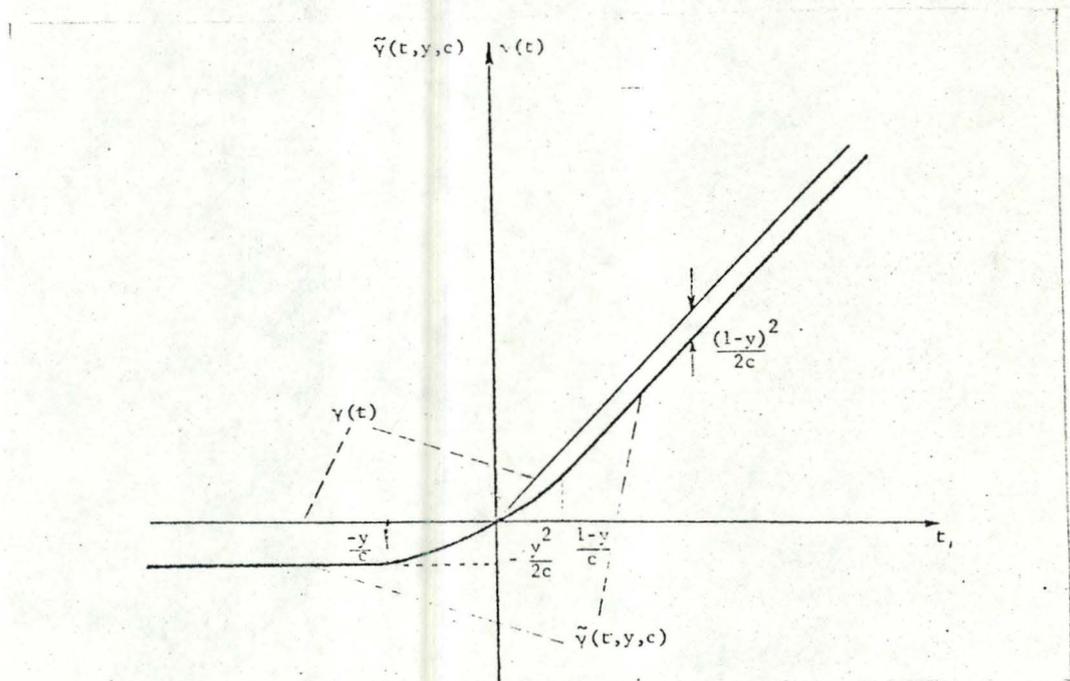
$$\tilde{\gamma}(t, y, c) = -\frac{y^2}{2c}$$

$$\gamma(t) = 0$$

$$\gamma(t) = \tilde{\gamma}(t, y, c) + \frac{y^2}{2c}$$

On peut donc rassembler les résultats des quatre cas en écrivant :

$$\gamma(t) \leq \tilde{\gamma}(t, y, c) + \frac{1}{2c} \max \{ y^2, (1-y)^2 \}$$



Puisque $y \leq 1$, de (6) on peut tirer :

$$\tilde{\gamma}(t, y, c) \leq \gamma(t) \leq \tilde{\gamma}(t, y, c) + \frac{1}{2c} \quad \text{pour tout } t \in \mathbb{R}$$

donc le paramètre c mesure la précision de l'approximation : plus c est grand, plus l'approximation est précise. Par contre, reprenant l'expression de $\tilde{\gamma}(t, y, c)$

$$\tilde{\gamma}(t, y, c) = \begin{cases} t - \frac{(1-y)^2}{2c} & \text{si } t \geq \frac{1-y}{c} \\ y t + \frac{c}{2} t^2 & \text{si } -\frac{y}{c} \leq t \leq \frac{1-y}{c} \\ -\frac{y^2}{2c} & \text{si } t \leq -\frac{y}{c} \end{cases}$$

on voit que y détermine si l'approximation est plus précise pour des valeurs positives ou négatives de t :

- Si y est proche de 1, l'approximation est plus précise pour des valeurs positives de t
pour $y = 1$ $\tilde{\gamma}(t, y, c) = \gamma(t)$ si $t \geq 0$
- Si y est proche de 0, l'approximation est plus précise pour des valeurs négatives de t
pour $y = 0$, $\tilde{\gamma}(t, y, c) = \gamma(t)$ si $t \leq 0$

Nous pourrions nous servir de ces remarques pour interpréter l'ajustement des paramètres y et c dans l'algorithme que nous allons considérer pour résoudre le problème (1).

d. Description de la procédure d'approximation pour résoudre le problème (1)

Pas 1

Etant donné les paramètres $c_k, y_k^i, i \in I$,
avec $c_k > 0, 0 \leq y_k^i \leq 1$,

remplacer chaque terme $\gamma [f_i(x)]$, $i \in I$, dans l'expression de la fonctionnelle g par $\tilde{\gamma} [f_i(x), y_k^i, c_k]$
pour obtenir une fonction \tilde{g}_k et résoudre le problème :

$$\begin{aligned} \min \quad & \tilde{g}_k(x) \\ & x \in Q \subset \mathbb{R}^n \end{aligned}$$

Pas 2

Si x_k est une solution de ce problème, faire un ajustement de c_k de telle façon que $c_{k+1} > c_k$

(par exemple, $c_{k+1} = \beta c_k$ avec $\beta > 1$)

et de $y_k^i, i \in I$ d'une certaine manière, pour obtenir y_{k+1}^i avec $0 \leq y_{k+1}^i \leq 1$

Pas 3

Faire $k = k+1$
et retourner au pas 1.

e. L'ajustement des paramètres $y_k^i, i \in I$

L'analyse de la méthode des multiplicateurs nous

a fourni une procédure d'ajustement des paramètres y_k^i qui augmentait d'une manière significative la vitesse de convergence de la méthode. On peut alors s'attendre, si on utilise une procédure analogue, à obtenir également des résultats avantageux. Ceci sera discuté plus loin. On considérera donc la formule d'ajustement suivante :

$$y_{k+1}^i = \begin{cases} 1 & \text{si } y_k^i + c_k f_i(x_k) \geq 1 \\ y_k^i + c_k f_i(x_k) & \text{si } 0 \leq y_k^i + c_k f_i(x_k) \leq 1 \\ 0 & \text{si } y_k^i + c_k f_i(x_k) \leq 0 \end{cases} \quad (7)$$

Interprétation

Supposons que x_k tend vers \bar{x} où \bar{x} est une solution optimale du problème (1), et que pour k assez grand on a

$$\frac{(1 - y)}{c_k} \leq f_i(x_k)$$

Ceci indique qu'on aura $f_i(\bar{x}) \geq 0$ et donc il vaudrait mieux qu'on approxime avec plus de précision $\mathcal{V}[f_i(x)]$ pour des valeurs positives de $f_i(x)$. D'après ce qu'on a remarqué dans le paragraphe précédent, on le fait en posant :

$$y_{k+1}^i = 1$$

De même si $f_i(x_k) \leq -\frac{y_k^i}{c_k}$

$f_i(\bar{x})$ sera négative et pour que l'approximation soit plus précise pour des valeurs négatives de $f_i(x_k)$, on pose

$$y_{k+1}^i = 0$$

Si

$$-\frac{y_k^i}{c_k} \leq f_i(x_k) \leq \frac{(1 - y_k^i)}{c_k}$$

on voudrait ajuster la précision de l'approximation de $\gamma(t)$ lorsque $f_i(x)$ passe de valeurs positives à des valeurs négatives ou vice-versa. En effet, $f_i(x_k)$ est proche de zéro, et peut changer de signe d'une étape à l'autre.

Si $f_i(x_k)$ est fort proche de zéro, l'itération

$$y_{k+1}^i = y_k^i + c_k f_i(x_k)$$

ne change pas beaucoup la valeur du paramètre y et on ne veut donc aucune précision spéciale de $\tilde{y}(t, y, c)$ pour des valeurs positives ou négatives de $f_i(x)$.

Mais si $f_i(x_k)$ devient "plus positive" par exemple, $f_i(x_k)$ se rapproche de $(1 - y_k^i)/c_k$ plutôt que de $-y_k^i/c_k$ et donc $f_i(x_k) + c_k f_i(x_k)$ se rapproche de 1.

L'approximation devient de plus en plus précise pour des valeurs positives de $f_i(x)$.

Si $f_i(x_k)$ devient négative, par contre, $f_i(x_k)$ se rapproche plutôt de $-\frac{y_k^i}{c_k}$ et donc

$f_i(x_k) + \frac{y_k^i}{c_k}$ se rapproche de zéro. L'itération

contribuera donc à rendre l'approximation de plus en plus précise pour les valeurs négatives de $f_i(x)$.

f. Présentation de l'analyse de la méthode proposée pour résoudre le problème (1).

La classe de problèmes de la forme (1) que nous avons considérée est fort vaste. Par exemple, s'y trouvent des problèmes de la forme :

$$\max \{ f_1(x), \dots, f_m(x) \}$$

En effet on peut montrer par récurrence que

$$\begin{aligned} \max \{ f_1(x), \dots, f_m(x) \} &= \\ &= f_1(x) + \gamma [f_2(x) - f_1(x) + \gamma [\dots \gamma [f_{m-1}(x) - f_{m-2}(x) \\ &\quad + \gamma [f_m(x) - f_{m-1}(x)]] \dots]] \end{aligned}$$

Puisque cette classe de problèmes est si variée, il est impossible de fournir une description et des notations valables pour tous les problèmes et ceci est un obstacle à l'analyse de la convergence pour le cas général.

Pour cette raison, on va se restreindre à la classe de problèmes suivants :

$$\begin{aligned} \min g [x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] \\ x \in Q \subset \mathbb{R}^n \end{aligned}$$

où g, f_1, \dots, f_m sont des fonctions continuellement différentiables.

Remarquons que cette classe de problèmes comprend celui de résoudre des systèmes d'équations de la forme :

$$h_i [x, \gamma [f_1(x)], \dots, \gamma [f_m(x)]] = 0 \quad i=1, \dots, m$$

au moyen du problème de minimisation :

$$\min \sum_{i=1}^m \left| h_i [x, \gamma [f_1(x)], \dots, \gamma [f_m(x)]] \right|^2$$

Voici comment nous allons procéder pour analyser la méthode :

Dans le chapitre I , on donnera les résultats de convergence pour notre méthode d'approximation.

Dans le chapitre II , on montrera qu'on peut obtenir des quantités qui jouent un rôle analogue à celui des multiplicateurs de Lagrange dans la minimisation avec contraintes. On verra aussi que nos résultats de convergence peuvent être utilisés pour obtenir des conditions d'optimalité pour la classe de problèmes que nous considérons.

Dans le chapitre III, on examinera la possibilité d'accélérer la convergence en utilisant l'itération (7). On clarifiera aussi la connexion de cette méthode avec la méthode des multiplicateurs.

CHAPITRE I

CONVERGENCE DE LA METHODE

Considérons le problème (9) , où l'on suppose que Q est non vide et que les fonctions :

$$g : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$$

$$f_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad i=1, \dots, m$$

sont partout continuellement différentiables .

Nous noterons $\nabla_x g$ le vecteur colonne des n premières dérivées de g .

Par contre, on notera $\frac{\partial g}{\partial t_i}$ $i=1, \dots, m$ les dérivées partielles de g par rapport au $(n+i)$ ème argument

$$\left(\frac{\partial g}{\partial t_i} = \frac{\partial g}{\partial y[f_i(x)]} \right).$$

Considérons maintenant le k ème problème de minimisation approximant :

$$\min_{x \in Q} g [x, \tilde{\gamma} [f_1(x), y_k^1, c_k], \dots, \tilde{\gamma} [f_m(x), y_k^m, c_k]] \quad (10)$$

$$\text{où } 0 < c_k < c_{k+1} \quad c_k \rightarrow \infty$$

$$0 \leq y_k^i \leq 1 \quad i=1, \dots, m \quad k=0, 1, \dots$$

et $\tilde{\gamma} [f_i(x), y_k^i, c_k]$ est donné par (6).

On peut utiliser une règle pour ajuster les y_k^i (par exemple y_k^i est gardé constant ou y_k^i est ajusté

par l'itération (7).)

Soit x_k une solution du problème (10) (en supposant qu'elle existe).

On a la proposition suivante :

Proposition 1

$$| g^* - g [x_k, \gamma [f_1(x_k)], \dots, \gamma [f_m(x_k)]] | \leq \frac{L}{c_k} ; k=0,1,\dots \quad (11)$$

$$\text{où } g^* = \inf_{x \in Q} g [x, \gamma [f_1(x)], \dots, \gamma [f_m(x)]] \quad (12)$$

$$L = \sum_{i=1}^m \sup_{(x, t_1, \dots, t_m) \in M} \left| \frac{\partial g}{\partial t_i} (x, t_1, \dots, t_m) \right| \quad (13)$$

$$M = \left\{ (x, t_1, \dots, t_m) : x \in Q, \gamma [f_i(x)] - \frac{1}{2} c_0 \leq t_i \leq \gamma [f_i(x)] \right\}_{i=1, \dots, m}$$

à condition que L soit fini.

Démonstration

Par la formule de Taylor on a que pour tout $x \in Q$

$$\left| g [x, \gamma [f_1(x)], \dots, \gamma [f_m(x)]] - g [x, \tilde{\gamma} [f_1(x), y_k^1, c_k], \dots, \tilde{\gamma} [f_m(x), y_k^m, c_k]] \right|$$

$$= \left| \begin{pmatrix} \gamma [f_1(x)] - \tilde{\gamma} [f_1(x), y_k^1, c_k] \\ \vdots \\ \gamma [f_m(x)] - \tilde{\gamma} [f_m(x), y_k^m, c_k] \end{pmatrix} \left(\frac{\partial g}{\partial t_1}, \dots, \frac{\partial g}{\partial t_m} \right) \Big|_{(x, \tilde{t}_1, \dots, \tilde{t}_m)} \right|$$

Donc , pour tout $x \in Q$, on a obtenu :

$$\begin{aligned} & \left| g(x, \gamma[f_1(x)] , \dots, \gamma[f_m(x)]) - g(x, \tilde{\gamma}[f_1(x), y_k^1, c_k], \dots \right. \\ & \qquad \qquad \qquad \left. \dots, \tilde{\gamma}[f_m(x), y_k^m, c_k]) \right| \\ & \leq \frac{1}{2c_k} \sum_{i=1}^m \sup_M \left| \frac{\partial g}{\partial t_i}(x, t_1, \dots, t_m) \right| = \frac{L}{2c_k} \quad (15) \end{aligned}$$

Ce qui entraîne également que :

$$\begin{aligned} & \left| \inf_{x \in Q} g(x, \gamma[f_1(x)] , \dots, \gamma[f_m(x)]) - \right. \\ & \qquad \qquad \qquad \left. \inf_{x \in Q} g[x, \tilde{\gamma}[f_1(x), y_k^1, c_k] , \dots, \tilde{\gamma}[f_m(x), y_k^m, c_k]] \right| \\ & \leq \frac{L}{2c_k} \end{aligned}$$

Puisque x_k est une solution optimale du problème (10), cette dernière inégalité peut s'écrire :

$$\left| g^* - g[x_k, \tilde{\gamma}[f_1(x_k), y_k^1, c_k] , \dots, \tilde{\gamma}[f_m(x_k), y_k^m, c_k]] \right| \leq \frac{L}{2c_k} \quad (16)$$

En combinant (15) et (16) on va enfin obtenir (11).
en effet :

$$\begin{aligned} & \left| g^* - g [x_k, \gamma[f_1(x_k)] , \dots, \gamma[f_m(x_k)]] \right| \\ & \leq \left| g^* - g[x_k, \tilde{\gamma}[f_1(x_k), y_k^1, c_k] , \dots, \tilde{\gamma}[f_m(x_k), y_k^m, c_k]] \right| + \\ & \qquad \qquad \qquad \left| g[x_k, \tilde{\gamma}[f_1(x_k), y_k^1, c_k] , \dots, \tilde{\gamma}[f_m(x_k), y_k^m, c_k]] - g[x_k, \gamma[f_1(x_k)] , \dots, \right. \\ & \qquad \qquad \qquad \left. \dots, \gamma[f_m(x_k)]] \right| \end{aligned}$$

$$\leq \frac{L}{2c_k} + \frac{L}{2c_k} = \frac{L}{c_k}$$

On peut déduire de cette proposition plusieurs corollaires, dont le suivant qui nous assure que la méthode considérée converge dans le cas où Q est un ensemble borné.

Corollaire 1.1

Soit Q un ensemble borné.

Alors

$$\lim_{k \rightarrow \infty} g[x_k, \gamma[f_1(x_k)], \dots, \gamma[f_m(x_k)]] = g^* \quad (17)$$

Démonstration

Puisque Q est borné, l'ensemble M de la proposition 1 est ensemble compact et comme les dérivées partielles de g par rapport aux t_i $i=1, \dots, m$ sont continues

$$\sup_{(x, t_1, \dots, t_m) \in M} \left| \frac{\partial g}{\partial t_i}(x, t_1, \dots, t_m) \right| \quad i=1, \dots, m$$

sont finis d'où L est fini.

Puisque c_k tend vers l'infini, en passant à la limite dans l'inégalité (11), on obtient :

$$\lim_{k \rightarrow \infty} g[x_k, \gamma[f_1(x_k)], \dots, \gamma[f_m(x_k)]] = g^*$$

Le corollaire suivant nous donne un résultat intéressant dans un cas particulier car il nous procure une borne à priori de l'erreur d'approximation.

Corollaire 1.2

Supposons que g a la forme particulière suivante :

$$g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] = g_0(x) + \sum_{i=1}^m \gamma[f_i(x)]$$

où $g_0 : \mathbb{R}^n \rightarrow \mathbb{R}$

Alors

$$\left| g^* - g[x_k, \gamma[f_1(x_k)], \dots, \gamma[f_m(x_k)]] \right| \leq \frac{m}{c_k}$$

Démonstration

Reprenons l'inégalité (a) de la proposition 1.

Pour tout $x \in Q$, on a :

$$\left| g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] - g[x, \tilde{\gamma}[f_1(x), y_k^1, c_k], \dots, \tilde{\gamma}[f_m(x), y_k^m, c_k]] \right|$$

$$\leq \frac{1}{2c_k} \sum_{i=1}^m \left| \frac{\partial g}{\partial t_i} \right| (x, \tilde{t}_1, \dots, \tilde{t}_m) \Big|$$

où $(x, \tilde{t}_1, \dots, \tilde{t}_m)$ est un point intermédiaire entre
 $(x, \gamma[f_1(x)], \dots, \gamma[f_m(x)])$ et $(x, \tilde{\gamma}[f_1(x), y_k^1, c_k], \dots,$
 $\dots, \tilde{\gamma}[f_m(x), y_k^m, c_k])$

or $\frac{\partial g}{\partial t_i}(x, t_1, \dots, t_m) = 1$ pour tout (x, t_1, \dots, t_m)

Ceci nous donne :

$$\left| g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] - g[x, \tilde{\gamma}[f_1(x), y_k^1, c_k], \dots, \tilde{\gamma}[f_m(x), y_k^m, c_k]] \right| \leq \frac{m}{2c_k}$$

et donc aussi :

$$\left| g^* - g[x_k, \tilde{\gamma}[f_1(x_k), y_k^1, c_k], \dots, \tilde{\gamma}[f_m(x_k), y_k^m, c_k]] \right| \leq \frac{m}{2c_k}$$

d'où

$$\begin{aligned} & \left| g^* - g[x_k, \gamma[f_1(x_k)], \dots, \gamma[f_m(x_k)]] \right| \\ & \leq \left| g^* - g[x_k, \tilde{\gamma}[f_1(x_k), y_k^1, c_k], \dots, \tilde{\gamma}[f_m(x_k), y_k^m, c_k]] \right| + \\ & \left| g[x_k, \tilde{\gamma}[f_1(x_k), y_k^1, c_k], \dots, \tilde{\gamma}[f_m(x_k), y_k^m, c_k]] - g[x_k, \gamma[f_1(x_k)], \dots, \gamma[f_m(x_k)]] \right| \\ & \leq \frac{m}{2c_k} + \frac{m}{2c_k} = \frac{m}{c_k} \end{aligned}$$

Le corollaire suivant prouve que chaque point d'accumulation de la suite $\{x_k\}$ est une solution du problème (9), dans le cas où Q est un ensemble fermé.

Corollaire 1.3

Soit \bar{x} un point d'accumulation de la suite $\{x_k\}$ et soit Q un ensemble fermé.
Alors

$$g[\bar{x}, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})]] = \min_{x \in Q} g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]]$$

Démonstration

Sans perte de généralité, supposons que la suite toute entière $\{x_k\}$ converge vers \bar{x} et soit S une boule fermée contenant la suite $\{x_k\}$.

Alors, on a que x_k est une solution optimale du problème :

$$\min_{x \in Q \cap S} g[x, \gamma[f_1(x), y_k^1, c_k], \dots, \gamma[f_m(x), y_k^m, c_k]]$$

et comme $Q \cap S$ est un ensemble borné, on peut appliquer le corollaire 1.1 et on obtient :

$$\inf_{x \in Q \cap S} g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] =$$

$$= \lim_{k \rightarrow \infty} g[x_k, \gamma[f_1(x_k)], \dots, \gamma[f_m(x_k)]]$$

Or , on a aussi que :

$$\begin{aligned} & \lim_{k \rightarrow \infty} g[x_k, \gamma[f_1(x_k)], \dots, \gamma[f_m(x_k)]] \\ = & g[\bar{x}, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})]] \end{aligned}$$

En rassemblant ces deux résultats on trouve :

$$\begin{aligned} g[\bar{x}, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})]] &= \inf_{x \in Q \cap S} g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] \\ &= \min_{x \in Q \cap S} g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] \end{aligned}$$

puisque , Q étant fermé et S compacte , $Q \cap S$ est compact.

Il reste à montrer que :

$$\begin{aligned} & \min_{x \in Q \cap S} g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] \\ = & \min_{x \in Q} g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] \end{aligned} \quad (18)$$

Mais comme S est une boule fermée contenant la suite $\{x_k\}$ arbitraire , on peut la prendre de rayon aussi grand que l'on veut et l'égalité (18) est donc vraie : le corollaire est démontré.

On a ainsi démontré que chaque point d'accumulation de la suite $\{x_k\}$ est un point de minimum du problème (9), mais rien ne dit qu'il existe un point d'accumulation.

D'autre part si Q est un ensemble compact, comme la fonction objective $g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]]$ est continue par rapport à x , le problème (9) possède au moins une solution. De plus chaque point de la suite $\{x_k\}$ est solution du problème :

$$\min_{x \in Q} g[x, \tilde{\gamma}[f_1(x), y_k^1, c_k], \dots, \tilde{\gamma}[f_m(x), y_k^m, c_k]]$$

Q étant fermé, $\{x_k\} \subset Q$ et comme Q est borné, cette suite possède au moins un point d'accumulation et par le corollaire que nous venons de voir, ce point est une solution du problème (9).

La proposition 1 et ses corollaires justifient donc l'emploi de la procédure d'approximation telle que nous l'avons construite, pour résoudre le problème (9).

Remarquons que pour prouver la convergence de la méthode, on a dû imposer aucune condition sur les paramètres y_k^i , $i=1, \dots, m$ et ceci nous laisse totalement libres de choisir une procédure d'ajustement qui accélérerait la convergence (voir chapitre III).

CHAPITRE II

CONVERGENCE DES MULTIPLICATEURSET CONDITIONS D'OPTIMALITE

Puisque la procédure d'approximation proposée est étroitement liée à la méthode des multiplicateurs, on s'attend à ce que le vecteur $y_k \in \mathbb{R}^m$, s'il est ajusté d'une manière appropriée (par une itération analogue à celle employée dans la méthode des multiplicateurs), converge vers un vecteur qui joue un rôle similaire à celui du vecteur optimal des multiplicateurs de Lagrange pour le problème différentiable avec contraintes.

Dans ce chapitre, nous verrons que ce vecteur des multiplicateurs existe effectivement et nous établirons les conditions d'optimalité pour le problème (9).

On supposera dans ce chapitre que Q est un ensemble convexe fermé.

Considérons à nouveau le problème approximant :

$$\min_{x \in Q} g[x, \tilde{\gamma}[f_1(x), y_k^1, c_k], \dots, \tilde{\gamma}[f_m(x), y_k^m, c_k]] \quad (19)$$

On a le résultat suivant :

Proposition 2

Soit Q un ensemble convexe fermé
 et soit (\bar{x}, \bar{y}) un point d'accumulation de la suite
 $\{(x_k, \tilde{y}_k)\}$ où $\tilde{y}_k = (\tilde{y}_k^1(x_k), \dots, \tilde{y}_k^m(x_k))'$ est défini
 de la manière suivante :

$$\tilde{y}_k^i(x_k) = \begin{cases} 1 & \text{si } 1 \leq y_k^i + c_k f_i(x_k) \\ y_k^i + c_k f_i(x_k) & \text{si } 0 \leq y_k^i + c_k f_i(x_k) \leq 1 \\ 0 & \text{si } y_k^i + c_k f_i(x_k) \leq 0 \end{cases}$$

Alors \bar{x} est une solution optimale du problème (9)
 et \bar{y} est un vecteur des multiplicateurs vérifiant :

$$\left\langle \nabla_x g + \sum_{i=1}^m \frac{\partial g}{\partial t_i} \bar{y}^i \nabla f_i \Big|_{x = \bar{x}}, (x - \bar{x}) \right\rangle \geq 0 \quad \text{pour tout } x \in Q$$

où l'on a adopté les notations du début du chapitre I,
 où $\langle \dots \rangle$ désigne le produit scalaire dans R^n
 et où

$$\begin{aligned} \bar{y}^i &= 0 & \text{si } f_i(\bar{x}) < 0 \\ \bar{y}^i &= 1 & \text{si } f_i(\bar{x}) > 0 \\ 0 \leq \bar{y}^i \leq 1 & & \text{si } f_i(\bar{x}) = 0 \end{aligned}$$

Démonstration

Calculons le gradient par rapport à x de la fonction
 objective du problème (19) :

$$\begin{aligned} & \nabla g[x, \tilde{\gamma}[f_1(x), y_k^1, c_k], \dots, \tilde{\gamma}[f_m(x), y_k^m, c_k]] \\ &= \nabla_x g + \sum_{i=1}^m \frac{\partial g}{\partial t_i} \nabla \tilde{\gamma}[f_i(x), y_k^i, c_k]_{|x} \end{aligned}$$

Par définition de $\tilde{\gamma}[f_i(x), y_k^i, c_k]$:

$$\nabla \tilde{\gamma}[f_i(x), y_k^i, c_k] = \begin{cases} \nabla f_i(x) & \text{si } 1 \leq y_k^i + c_k f_i(x) \\ [y_k^i + c_k f_i(x)] \nabla f_i(x) & \text{si } 0 \leq y_k^i + c_k f_i(x) \leq 1 \\ 0 & \text{si } y_k^i + c_k f_i(x) \leq 0 \end{cases}$$

Le gradient par rapport à x de la fonction objective peut alors s'écrire :

$$\nabla_x g + \sum_{i=1}^m \frac{\partial g}{\partial t_i} \tilde{y}_k^i(x) \nabla f_i|_x \quad (20)$$

où $\tilde{y}_k^i(x)$, $i=1, \dots, m$ est donné par :

$$\tilde{y}_k^i(x) = \begin{cases} 1 & \text{si } 1 \leq y_k^i + c_k f_i(x) \\ y_k^i + c_k f_i(x) & \text{si } 0 \leq y_k^i + c_k f_i(x) \leq 1 \\ 0 & \text{si } y_k^i + c_k f_i(x) \leq 0 \end{cases} \quad (21)$$

Puisque Q est un ensemble convexe et que x_k est une solution optimale du problème (19), on a la condition nécessaire :

$$\left\langle \nabla_x g + \sum_{i=1}^m \frac{\partial g}{\partial t_i} \bar{y}_k^i(x_k) \nabla f_i \Big|_{x=x_k}, (x-x_k) \right\rangle \geq 0 \quad (22)$$

pour tout $x \in Q$

Soit $\{x_k\}_{k \in K}$, une sous-suite de $\{x_k\}$ qui converge vers \bar{x} .

Puisque, par hypothèse, Q est fermé, d'après le corollaire 1.3, \bar{x} est une solution optimale du problème (9).

y_k étant une fonction continue de x , la suite $\{y_k\}_{k \in K}$

$$\{y_k\}_{k \in K} = \{(y_k^1(x_k), \dots, y_k^m(x_k))\}_{k \in K}$$

a aussi une limite $\bar{y} = (\bar{y}^1, \dots, \bar{y}^m)$,

$$\bar{y}^i = \bar{y}^i(\bar{x}) = \begin{cases} 1 & \text{si } f_i(\bar{x}) > 0 \\ 0 \leq \bar{y}^i(\bar{x}) \leq 1 & \text{si } f_i(\bar{x}) = 0 \\ 0 & \text{si } f_i(\bar{x}) < 0 \end{cases}$$

En "passant à la limite" dans (22), on a que (\bar{x}, \bar{y}) doit vérifier :

$$\left\langle \nabla_x g + \sum_{i=1}^m \frac{\partial g}{\partial t_i} \bar{y}^i \nabla f_i \Big|_{x=\bar{x}}, (x-\bar{x}) \right\rangle \geq 0$$

pour tout $x \in Q$

Maintenant, avant d'établir les conditions nécessaires d'optimalités, définissons les notions de minimum local et de minimum local isolé pour le problème (9).

Définition

\bar{x} est appelé minimum local pour le problème (9) si \bar{x} minimise $g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]]$ sur un ensemble de la forme $Q \cap \{x : \|x - \bar{x}\| \leq \varepsilon\}$ où $\|\cdot\|$ désigne la norme euclidienne et $\varepsilon > 0$ est un scalaire

\bar{x} est appelé minimum local isolé pour le problème (9) si \bar{x} est un point de minimum unique de $g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]]$ sur $Q \cap \{x : \|x - \bar{x}\| \leq \varepsilon\}$

Proposition 3

(condition nécessaire d'optimalité pour le problème (9))

Soit Q un ensemble convexe fermé et \bar{x} un point de minimum local isolé pour le problème (9).

Alors il existe un vecteur des multiplicateurs $\bar{y} = (\bar{y}^1, \dots, \bar{y}^m)'$ vérifiant :

$$\left\langle \nabla_x g + \sum_{i=1}^m \frac{\partial g}{\partial t_i} \bar{y}^i \nabla f_i \Big|_{x = \bar{x}}, (x - \bar{x}) \right\rangle \geq 0$$

pour tout $x \in Q$

$$\bar{y}^i = 0 \quad \text{si } f_i(\bar{x}) < 0$$

$$\bar{y}^i = 1 \quad \text{si } f_i(\bar{x}) > 0$$

$$0 \leq \bar{y}^i \leq 1 \quad \text{si } f_i(\bar{x}) = 0$$

Démonstration

Soit \bar{Q} un ensemble de la forme $Q \cap \{x : \|x - \bar{x}\| \leq \varepsilon\}$ contenant \bar{x} , un point de minimum unique de g sur \bar{Q} . Considérons la procédure d'approximation pour le problème suivant :

$$\min_{x \in \bar{Q}} g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]]$$

puisque \bar{Q} est compact, la suite $\{x_k\}$ est bien définie et possède au moins un point d'accumulation, qui d'après le corollaire 1.3 est une solution optimale du problème ci-dessus. Puisque celle-ci est unique, on a que la suite $\{x_k\}$ converge vers \bar{x} .

Considérons maintenant la suite $\{\tilde{y}_k\}$ où \tilde{y}_k est défini comme dans la proposition 2. Puisque $\{x_k\}$ converge la suite $\{y_k\}$ a également une limite \bar{y} . On peut maintenant appliquer la proposition 2: (\bar{x}, \bar{y}) satisfait :

$$\left\langle \nabla_x g + \sum_{i=1}^m \frac{\partial g}{\partial t_i} \bar{y}^i \nabla f_i \Big|_{x=\bar{x}}, (x-\bar{x}) \right\rangle \geq 0$$

pour tout $x \in Q$

$$\bar{y}^i = 1 \quad \text{si } f_i(\bar{x}) > 0$$

$$\bar{y}^i = 0 \quad \text{si } f_i(\bar{x}) < 0$$

$$0 \leq \bar{y}^i \leq 1 \quad \text{si } f_i(\bar{x}) = 0$$

Remarque

Lorsque $Q = \mathbb{R}^n$, la condition nécessaire ci-dessus donne :

$$\nabla_x g + \sum_{i=1}^m \frac{\partial g}{\partial t_i} \bar{y}^i \nabla f_i \Big|_{x = \bar{x}} = 0$$

Cette remarque peut être utile en pratique car on peut l'utiliser dans un programme comme critère d'arrêt.

CHAPITRE III

L'ITERATION DES MULTIPLICATEURS ACCELERE
LA CONVERGENCE

Dans l'introduction, nous avons déjà donné une formule d'ajustement des multiplicateurs (voir (7)) qui est analogue à celle employée dans la méthode des multiplicateurs. Nous en avons déjà donné une interprétation. Maintenant, nous allons voir que , comme dans la méthode des multiplicateurs, cette méthode d'ajustement est très avantageuse car elle accélère la convergence.

Pour le faire, nous envisagerons d'abord un cas particulier du problème (9) , puis nous ferons deux généralisations.

Nous nous intéressons donc à la formule d'ajustement des multiplicateurs suivante :

$$y_{k+1}^i = \begin{cases} 1 & \text{si } 1 \leq y_k^i + c_k f_i(x_k) \\ y_k^i + c_k f_i(x_k) & \text{si } 0 \leq y_k^i + c_k f_i(x_k) \leq 1 \\ 0 & \text{si } y_k^i + c_k f_i(x_k) \leq 0 \end{cases} \quad (23)$$

a.) Considérons d'abord le cas particulier du problème (9)

$$\min_x \left\{ g_0(x) + \sum_{i=1}^m a_i \gamma[f_i(x)] \right\} = \min_x \left\{ g_0(x) + \sum_{i=1}^m \delta[a_i f_i(x)] \right\}$$

où a_i , $i=1, \dots, m$ sont des scalaires strictement positifs.

En introduisant des variables supplémentaires z_1, \dots, z_m , ce problème est équivalent au problème :

$$\begin{aligned} \min_{x, z} \quad & \left\{ g_0(x) + \sum_{i=1}^m z_i \right\} & (24) \\ a_i f_i(x) & \leq z_i \\ 0 & \leq z_i \end{aligned}$$

qu'on peut résoudre par la méthode des multiplicateurs, c'est-à-dire minimiser par rapport à x et z le problème suivant :

$$\begin{aligned} \min \quad & \left\{ g_0(x) + \sum_{i=1}^m z_i + \right. \\ & \left. + \frac{1}{2c_k} \sum_{i=1}^m \left[\left[\max \left\{ 0, y_k^i + c_k (a_i f_i(x) - z_i) \right\} \right]^2 - (y_k^i)^2 \right] \right\} & (25) \end{aligned}$$

pour une suite croissante $\{c_k\}$ de scalaires strictement positifs et une suite $\{y_k\}$, $y_k \in \mathbb{R}^m$, où y_k est ajusté par l'itération suivante :

$$\begin{cases} y_{k+1}^i = y_k^i + c_k (a_i f_i(x_k) - z_{i_k}) & \text{si } y_k^i + c_k (a_i f_i(x_k) - z_{i_k}) \geq 0 \\ y_{k+1}^i = 0 & \text{sinon} \end{cases} \quad (26)$$

où x_k , z_{i_k} , $i=1, \dots, m$ est solution optimale du problème (25)^k

Maintenant par une procédure analogue à celle employée dans le paragraphe b de l'introduction, on peut minimiser la fonction objective du problème (25) par rapport à z et le problème (25) se réduit au problème :

$$\begin{aligned} & \min_x \left\{ g_0(x) + \sum_{i=1}^m \tilde{\gamma} [a_i f_i(x), y_k^i, c_k] \right\} \\ & = \min_x g[x, \tilde{\gamma} [a_1 f_1(x), y_k^1, c_k], \dots, \tilde{\gamma} [a_m f_m(x), y_k^m, c_k]] \quad (27) \end{aligned}$$

qui est le problème approximant de la procédure que nous étudions.

Voyons maintenant comment se transforme l'itération (26). Si x_k est une solution du problème approximant (27), on a trois cas :

$$\text{1^{er} cas : si } y_k^i + c_k a_i f_i(x_k) \geq 0 \text{ et } a_i f_i(x_k) - \frac{(1-y_k^i)}{c_k} \leq 0$$

ce qui est équivalent à avoir

$$0 \leq y_k^i + c_k a_i f_i(x_k) \leq 1$$

dans ce cas on a vu que la solution de la minimisation de

$$z_i + \frac{1}{2c_k} \left[\left[\max \left\{ 0, y_k^i + c_k (a_i f_i(x_k) - z_i) \right\} \right]^2 - (y_k^i)^2 \right] \quad (28)$$

par rapport à z_i était $z_i = 0$

Dans ce cas, l'itération (26) donne :

$$y_{k+1}^i = y_k^i + c_k a_i f_i(x_k) \text{ puisque } y_k^i + c_k a_i f_i(x_k) \geq 0$$

$$\underline{2^{\text{ème}} \text{ cas}} : y_k^i + c_k a_i f_i(x_k) \geq 1$$

Dans ce cas, la solution de la minimisation de (28) par rapport à z_i est :

$$z_i = a_i f_i(x_k) - \frac{(1 - y_k^i)}{c_k}$$

et l'itération (26) donne :

$$\begin{aligned} y_{k+1}^i &= \max \left\{ 0, y_k^i + c_k (a_i f_i(x_k) - z_i) \right\} \\ &= \max \left\{ 0, y_k^i + c_k \left[a_i f_i(x_k) - a_i f_i(x_k) + \frac{(1 - y_k^i)}{c_k} \right] \right\} \\ &= \max \{ 0, 1 \} = 1 \end{aligned}$$

$$\underline{3^{\text{ème}} \text{ cas}} \quad y_k^i + c_k a_i f_i(x_k) \leq 0$$

La solution de la minimisation de (28) par rapport à z_i était :

$$z_i = 0$$

et l'itération (26) donne , puisque $y_k^i + c_k a_i f_i(x_k) \leq 0$:

$$y_{k+1}^i = 0$$

En rassemblant les résultats des trois cas , on obtient que l'itération (26) donne l'itération :

$$y_{k+1}^i = \begin{cases} 1 & \text{si } y_k^i + c_k a_i f_i(x_k) \geq 1 \\ y_k^i + c_k a_i f_i(x_k) & \text{si } 0 \leq y_k^i + c_k a_i f_i(x_k) \leq 1 \\ 0 & \text{si } y_k^i + c_k a_i f_i(x_k) \leq 0 \end{cases}$$

qui est identique à l'itération (23) que nous étudions.

Donc la méthode d'approximation avec l'itération (23) est équivalente à la méthode des multiplicateurs dans le cas où le problème (9) a la forme particulière que nous avons considérée ici. Donc, comme l'itération des multiplicateurs (26) accélère la convergence et que pour cette itération, on a pas besoin de faire croître c_k vers l'infini pour que la méthode converge, on gardera ces avantages en ajustant les multiplicateurs par l'itération (23) dans la procédure d'approximation pour le problème particulier considéré.

b.) Première généralisation

Supposons que $Q = \mathbb{R}^n$.

Considérons le cas particulier du problème (9) où la fonction objective a la forme suivante :

$$g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] = g_0(x) + \sum_{i=1}^m g_i(x) \gamma[f_i(x)] \quad (29)$$

où $g_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad i=1, \dots, m$

Supposons que \bar{x} est un minimum local isolé du problème (29) dans une boule ouverte $B(\bar{x}, \epsilon)$

Nous définissons :

$$I^+ = \{ i : f_i(\bar{x}) > 0, i=1, \dots, m \}$$

$$I^- = \{ i : f_i(\bar{x}) < 0, i=1, \dots, m \}$$

$$I^0 = \{ i : f_i(\bar{x}) = 0, i=1, \dots, m \}$$

Supposons que ε est suffisamment petit pour assurer que

$$f_i(x) > 0 \quad \text{pour tout } x \in B(\bar{x}, \varepsilon) \quad i \in I^+$$

$$f_i(x) < 0 \quad \text{pour tout } x \in B(\bar{x}, \varepsilon) \quad i \in I^-$$

Supposons aussi que $g_i(\bar{x}) \neq 0$ pour tout $i \in I^0$. (30)

Il existe un voisinage V de \bar{x} inclu dans $B(\bar{x}, \varepsilon)$ pour lequel le problème (29) peut s'écrire localement :

$$\min_x \left\{ g_0(x) + \sum_{i \in I^+} g_i(x) f_i(x) + \sum_{i \in I^{0+}} \max \{ 0, g_i(x) f_i(x) \} \right. \\ \left. + \sum_{i \in I^{0-}} \min \{ 0, g_i(x) f_i(x) \} \right\} \quad (31)$$

$$\text{où } I^{0+} = \{ i : g_i(\bar{x}) > 0, f_i(\bar{x}) = 0 \}$$

$$I^{0-} = \{ i : g_i(\bar{x}) < 0, f_i(\bar{x}) = 0 \}$$

En effet, si $i \in I^-$, $g_i(x) \vee [f_i(x)] = g_i(x) \cdot 0 = 0$

pour tout $x \in B(\bar{x}, \varepsilon)$

si $i \in I^+$, $g_i(x) \vee [f_i(x)] = g_i(x) f_i(x)$

pour tout $x \in B(\bar{x}, \varepsilon)$

et il existe un voisinage V de \bar{x} inclu dans $B(\bar{x}, \varepsilon)$ tel que

$g_i(x) > 0$ pour tout $x \in V$, $i \in I^{0+}$

$g_i(x) < 0$ pour tout $x \in V$, $i \in I^{0-}$

donc

pour $i \in I^{0+}$ $g_i(x) \vee [f_i(x)] = g_i(x) f_i(x)$ si $f_i(x) g_i(x) \geq 0$

$= g_i(x) \cdot 0$ si $f_i(x) g_i(x) \leq 0$

$$\text{d'où } g_i(x) \gamma[f_i(x)] = \max \{0, f_i(x)g_i(x)\}$$

$$\begin{aligned} \text{pour } i \in I^{0-} \quad g_i(x) \gamma[f_i(x)] &= g_i(x) f_i(x) \text{ si } f_i(x)g_i(x) \leq 0 \\ &= g_i(x) \cdot 0 \text{ si } f_i(x)g_i(x) \geq 0 \end{aligned}$$

$$\text{d'où } g_i(x) \gamma[f_i(x)] = \min \{0, f_i(x)g_i(x)\}$$

Le problème (31) est encore équivalent au problème :

$$\min_{x, z} \left\{ g_0(x) + \sum_{i \in I^+} g_i(x) f_i(x) + \sum_{i \in I^{0+}} z_i + \sum_{i \in I^{0-}} \min\{0, g_i(x) f_i(x)\} \right\} \quad (32)$$

$$\begin{aligned} &g_i(x) f_i(x) \leq z_i \\ &0 \leq z_i, i \in I^{0+} \end{aligned}$$

Remarquons que comme pour $i \in I^{0-}$:

$$\min\{0, g_i(x) f_i(x)\} \leq g_i(x) f_i(x)$$

et

$$\min\{0, g_i(x) f_i(x)\} \leq 0$$

on a que :

$$\sum_{i \in I^{0-}} \min\{0, g_i(x) f_i(x)\} \leq \sum_{i \in \tilde{I}^{0-}} g_i(x) f_i(x)$$

pour tout sous-ensemble $\tilde{I}^{0-} \subset I^{0-}$

De plus ,

$$\sum_{i \in I^{0-}} \min\{0, g_i(\bar{x}) f_i(\bar{x})\} = \sum_{i \in I^{0-}} g_i(\bar{x}) f_i(\bar{x}) = 0$$

Alors, on peut conclure que \bar{x} est également une solu-

tion optimale du problème :

$$\min \left\{ g_0(x) + \sum_{i \in I^+} g_i(x) f_i(x) + \sum_{i \in I^{0+}} z_i + \sum_{i \in \tilde{I}^{0-}} g_i(x) f_i(x) \right\} \quad (33)$$

$$g_i(x) f_i(x) \leq z_i$$

$$0 \leq z_i, \quad i \in I^{0+}$$

pour tout $\tilde{I}^{0-} \subset I^{0-}$.

Si on fait l'hypothèse que

$$\nabla [g_i(x) f_i(x)] \Big|_{x=\bar{x}}, \quad i \in I^0 \quad (34)$$

sont des vecteurs linéairement indépendants, alors I^{0-} est vide.

En effet, puisque \bar{x} est une solution du problème (33), il doit satisfaire la condition nécessaire de Kuhn-Tucker :

$$\nabla g_0(x) + \sum_{i \in I^+} \nabla [g_i(x) f_i(x)] \Big|_{x=\bar{x}} + \sum_{i \in \tilde{I}^{0-}} \nabla [g_i(x) f_i(x)] \Big|_{x=\bar{x}} + \sum_{i \in I^{0+}} \bar{\lambda}^i \nabla [g_i(x) f_i(x)] \Big|_{x=\bar{x}} = 0 \quad (35)$$

pour tout sous-ensemble $\tilde{I}^{0-} \subset I^{0-}$

où $\bar{\lambda}^i$ est le multiplicateur optimal de Lagrange pour la contrainte $g_i(x) f_i(x) - z_i \leq 0$, $i \in I^{0+}$

Puisque les $\nabla [g_i(x) f_i(x)] \Big|_{x=\bar{x}}$, $i \in I^0$ sont linéairement indépendants, l'égalité (35) entraîne que

tous les sous-ensemble \tilde{I}^{0-} de I^{0-} sont vides et donc que I^{0-} est vide.

Finalement, le problème (29) dans un voisinage V de \bar{x} est équivalent au problème :

$$\min_{\substack{g_i(x)f_i(x) \leq z_i \\ 0 \leq z_i, i \in I^0}} \left\{ g_0(x) + \sum_{i \in I^+} g_i(x)f_i(x) + \sum_{i \in I^0} z_i \right\} \quad (36)$$

Nous allons résoudre ce problème par une méthode analogue à celle des multiplicateurs mais où le paramètre de pénalisation dépend de x . Cette méthode possède les mêmes avantages que la méthode des multiplicateurs que nous avons étudiée dans la première partie (pour la description et l'analyse de cette nouvelle méthode, voir appendice C).

Cette méthode, appliquée au problème (36) consiste à trouver x_k et $z_i, i \in I^0$ résolvant le problème :

$$\min_{x, z_i} \left\{ g_0(x) + \sum_{i \in I^+} g_i(x)f_i(x) + \sum_{i \in I^0} z_i + \sum_{i \in I^0} \frac{1}{2c_k^i(x)} \left[\max \left\{ 0, y_k^i + \tilde{c}_k^i(x) [g_i(x)f_i(x) - z_i] \right\}^2 - (y_k^i)^2 \right] \right\} \quad (43)$$

où on a pris $\tilde{c}_k^i(x) = \frac{c_k}{g_i(x)}$

et si $x_k, z_i, i \in I^0$ est une solution du problème (43), poser

$$y_{k+1}^i = \max \left\{ 0, y_k^i + \tilde{c}_k^i(x_k) [g_i(x_k) f_i(x_k) - z_i] \right\} \quad i \in I^0$$

$$c_{k+1} \geq c_k$$

Minimisons d'abord (43) par rapport à z_i $i \in I^0$

Soit $i \in I^0$, $x \in B(\bar{x}, \varepsilon)$ fixés et cherchons le minimum par rapport à z_i du terme :

$$\min_{z_i \geq 0} \left\{ z_i + \frac{1}{2c_k^i(x)} \left[\max \left\{ 0, y_k^i + \tilde{c}_k^i(x) [g_i(x) f_i(x) - z_i] \right\}^2 - y_k^{i2} \right] \right\} \quad (44)$$

1) supposons d'abord que $y_k^i + \tilde{c}_k^i(x) g_i(x) f_i(x) \leq 0$

$$\text{donc } y_k^i + c_k f_i(x) \leq 0$$

alors pour tout $z_i \geq 0$

$$\max \left\{ 0, y_k^i + \tilde{c}_k^i(x) [g_i(x) f_i(x) - z_i] \right\} = 0 \quad (45)$$

et la solution du problème (44) est $z_i = 0$

et le terme correspondant à $i \in I^0$ du problème (43) se réduit à :

$$-\frac{1}{2} \frac{(y_k^i)^2}{\tilde{c}_k^i(x)} = g_i(x) \left(-\frac{1}{2} \frac{(y_k^i)^2}{c_k} \right)$$

2) Supposons maintenant que $y_k^i + \tilde{c}_k^i(x) g_i(x) f_i(x) \geq 0$

$$\text{ou encore que } y_k^i + c_k f_i(x) \geq 0$$

Cherchons d'abord le minimum par rapport à z_i , $z_i \geq 0$ de

$$z_i + \frac{1}{2\tilde{c}_k^i(x)} \left[\left[y_k^i + \tilde{c}_k^i(x) [g_i(x)f_i(x) - z_i] \right]^2 - (y_k^i)^2 \right]$$

Dérivons cette expression par rapport à z_i et annulons cette dérivée :

$$1 + \tilde{c}_k^i(x) z_i - \tilde{c}_k^i(x) g_i(x) f_i(x) - y_k^i = 0$$

$$z_i = \frac{-1 + y_k^i + \tilde{c}_k^i(x) g_i(x) f_i(x)}{\tilde{c}_k^i(x)}$$

On trouve donc comme solution du problème (44) :

$$* \text{ si } y_k^i + \tilde{c}_k^i(x) g_i(x) f_i(x) \geq 1$$

$$z_i = \frac{y_k^i - 1}{\tilde{c}_k^i(x)} + g_i(x) f_i(x)$$

et dans ce cas :

$$\max \left\{ 0, y_k^i + \tilde{c}_k^i(x) \left[f_i(x) g_i(x) - \frac{y_k^i - 1 + \tilde{c}_k^i(x) g_i(x) f_i(x)}{\tilde{c}_k^i(x)} \right] \right\}$$

$$= \max \left\{ 0, y_k^i + \tilde{c}_k^i(x) f_i(x) g_i(x) - y_k^i + 1 - \tilde{c}_k^i(x) g_i(x) f_i(x) \right\}$$

$$= \max \{ 0, 1 \} = 1$$

(47)

et donc le terme correspondant à $i \in I^0$ dans l'expression

du problème (43) se réduit à :

$$\begin{aligned} & \frac{y_k^i - 1 + \tilde{c}_k^i(x)g_i(x)f_i(x)}{\tilde{c}_k^i(x)} + \frac{1}{2\tilde{c}_k^i(x)} (1 - y_k^i) \\ &= \frac{2y_k^i - 2 + 2\tilde{c}_k^i(x)g_i(x)f_i(x) + 1 - y_k^i}{2\tilde{c}_k^i(x)} \\ &= g_i(x)f_i(x) - g_i(x) \frac{(1 - y_k^i)^2}{2c_k} \\ &= g_i(x) \left[f_i(x) - \frac{(1 - y_k^i)^2}{2c_k} \right] \end{aligned}$$

* par contre si $0 \leq y_k^i + c_k^i(x)g_i(x)f_i(x) \leq 1$

alors c'est $z_i = 0$ qui est solution du problème (44) et

$$\begin{aligned} & \max \left\{ 0, y_k^i + \tilde{c}_k^i(x) [g_i(x)f_i(x) - z_i] \right\} \\ &= y_k^i + \tilde{c}_k^i(x)g_i(x)f_i(x) \end{aligned}$$

et le terme correspondant à $i \in I^0$ dans l'expression du problème (43) se réduit à :

$$\begin{aligned} & \frac{1}{2\tilde{c}_k^i(x)} \left[\left[y_k^i + \tilde{c}_k^i(x)g_i(x)f_i(x) \right]^2 - \left[y_k^i \right]^2 \right] \\ &= \frac{1}{2\tilde{c}_k^i(x)} \left[\left(y_k^i \right)^2 + \left[\tilde{c}_k^i(x)g_i(x)f_i(x) \right]^2 + 2y_k^i\tilde{c}_k^i(x)g_i(x)f_i(x) - \left(y_k^i \right)^2 \right] \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \tilde{c}_k^i(x) [g_i(x) f_i(x)]^2 + y_k^i \tilde{c}_k^i(x) f_i(x) g_i(x) \\
&= \frac{1}{2} c_k g_i(x) [f_i(x)]^2 + y_k^i f_i(x) g_i(x) \\
&= g_i(x) \left[\frac{c_k}{2} [f_i(x)]^2 + y_k^i f_i(x) \right]
\end{aligned}$$

Se rappelant l'expression de $\tilde{\gamma}[f_i(x), y_k^i, c_k]$ (voir(6)), le problème (43) peut s'écrire :

$$\min_x g_0(x) + \sum_{i \in I^+} g_i(x) f_i(x) + \sum_{i \in I^0} g_i(x) \tilde{\gamma}[f_i(x), y_k^i, c_k] \quad (48)$$

Maintenant voyons ce que devient l'itération des multiplicateurs si x_k est une solution optimale du problème (48).

$$y_{k+1}^i = \max\{0, y_k^i + \tilde{c}_k^i(x_k) [g_i(x_k) f_i(x_k) - z_i]\} \quad i \in I^0$$

$$1) \text{ si } y_k^i + \tilde{c}_k^i(x_k) g_i(x_k) f_i(x_k) = y_k^i + c_k f_i(x_k) \leq 0$$

$$\text{d'après (45) on a } y_{k+1}^i = 0$$

$$2) \text{ si } 1 \leq y_k^i + c_k f_i(x_k)$$

$$\text{d'après (46) on a } y_{k+1}^i = 1$$

$$3) \text{ } 0 \leq y_k^i + c_k f_i(x_k) \leq 1$$

$$\text{d'après (47) on a } y_{k+1}^i = y_k^i + c_k f_i(x_k)$$

Maintenant, on voit clairement que si on est assez proche de \bar{x} , résoudre le problème (29) par la procédure d'approximation et la méthode d'ajustement des multiplicateurs utilisant l'itération (23) revient à résoudre le problème équivalent (36) par la méthode des multiplicateurs où le paramètre de pénalisation dépend de x .

D'après ce que nous avons vu sur cette dernière méthode (voir appendice C) on a donc de grands avantages à utiliser cette itération (23) puisqu'elle accélère la convergence et qu'on a pas besoin de faire croître c_k vers l'infini, évitant ainsi les problèmes d'instabilité numérique.

Tout ceci est valable uniquement pour des problèmes de la forme (29). Nous allons maintenant faire une dernière généralisation, montrant que l'analyse de ce chapitre est également valable pour le cas général du problème (9).

c.) Deuxième généralisation

Considérons le problème (9)

$$\min_x g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]]$$

où g et f_i $i=1, \dots, m$ sont des fonctions différentiables

Pour faciliter les calculs et sans perte de généralité, supposons que $I^0 = \{1, \dots, m\}$

Développons la fonction objective autour de

$$\bar{t} = (\gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})])$$

$$\begin{aligned}
G(x) &= g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] \\
&= g[x, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})]] + \\
&\quad + \sum_{i=1}^m \frac{\partial g}{\partial t_i} [x, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})]] \cdot \gamma[f_i(x)] + \\
&\quad + O_x \left(\sum_{i=1}^m \gamma[f_i(x)] \right) \tag{49}
\end{aligned}$$

où O_x est une fonction de $x, \gamma[f_i(x)]$ $i=1, \dots, m$ telle que pour tout x :

$$\lim_{\sum_{i=1}^m \gamma[f_i(x)] \rightarrow 0} \frac{O_x \left(\sum_{i=1}^m \gamma[f_i(x)] \right)}{\sum_{i=1}^m \gamma[f_i(x)]} = 0 \tag{50}$$

Maintenant, remarquons que la fonction (49) est de la forme de la fonction objective du problème (29) sauf des termes d'ordre plus grand ou égal à 2. On s'attend à ce que ces termes supplémentaires soient négligeables si on est assez proche de \bar{x} et qu'ils tendent vers 0 lorsque x_k tend vers \bar{x} . Ainsi, tout ce que nous avons dit dans le paragraphe précédent serait valable asymptotiquement pour le cas général.

C'est ce que nous allons voir un peu plus en détail. Considérons la fonction $\tilde{G}(x)$ du type de la fonction objective du problème (29) du paragraphe précédent :

$$\begin{aligned}
\tilde{G}(x) &= g[x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] + \\
&\quad + \sum_{i=1}^m \frac{\partial g}{\partial t_i} [x, \gamma[f_1(x)], \dots, \gamma[f_m(x)]] \gamma[f_i(x)]
\end{aligned}$$

Puisque \bar{x} est une solution optimale du problème (49) on a d'après la proposition 3 du chapitre précédent, la condition nécessaire :

Il existe un vecteur $\bar{y} = (\bar{y}^1, \dots, \bar{y}^m)'$; avec $0 \leq \bar{y}^i \leq 1$ tel que :

$$\nabla_x g|_{x=\bar{x}} + \sum_{i=1}^m \frac{\partial g}{\partial t_i} \bar{y}^i \nabla f_i|_{x=\bar{x}} = 0 \quad (51)$$

L'expression (51) ci-dessus peut encore s'écrire :

$$\nabla_x g[\bar{x}, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})]] + \sum_{i=1}^m \frac{\partial g}{\partial t_i} [\bar{x}, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})]] \bar{y}^i \nabla f_i(\bar{x}) = 0 \quad (52)$$

Remarquons maintenant que $\tilde{G}(x)$ est aussi une fonction de $\gamma[f_i(x)]$ vérifiant également la condition (51) En effet :

$$\begin{aligned} & \nabla_x \tilde{G}|_{x=\bar{x}} + \sum_{i=1}^m \frac{\partial \tilde{G}}{\partial t_i} \bar{y}^i \nabla f_i|_{x=\bar{x}} = \\ & = \nabla_x [g(x, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})])]|_{x=\bar{x}} + \\ & + \sum_{i=1}^m \frac{\partial g}{\partial t_i} [x, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})]]|_{x=\bar{x}} \gamma[f_i(x)]|_{x=\bar{x}} + \\ & + \sum_{i=1}^m \frac{\partial g}{\partial t_i} [x, \gamma[f_1(\bar{x})], \dots, \gamma[f_m(\bar{x})]] \bar{y}^i \nabla f_i|_{x=\bar{x}} \\ & = 0 \quad \text{d'après (52)} \end{aligned}$$

Supposons alors que \bar{x} est aussi une solution optimale du problème :

$$\min \tilde{G}(x) \quad (53)$$

(Cette hypothèse n'est évidemment pas toujours vérifiée mais a lieu cependant dans la plupart des cas.)

Si nous utilisons la procédure d'approximation pour minimiser $G(x)$, alors à l'étape k , le gradient de la fonction objective approximante sera nul au point x_k .

Puisque

$$G(x) = \tilde{G}(x) + O_x \left(\sum_{i=1}^m \gamma[f_i(x)] \right)$$

alors le gradient de la fonction approximante du problème (53) sera égal à :

$$-\nabla_x O_{x_k} \left(\sum_{i=1}^m \tilde{\gamma}[f_i(x_k), y_k^i, c_k] \right)$$

Comme O_x ne contient que des termes d'ordre 2 ou d'ordre plus grand et que $x_k \rightarrow \bar{x}$, $y_k \rightarrow \bar{y}$

$$\nabla_x O_{x_k} \left(\sum_{i=1}^m \tilde{\gamma}[f_i(x_k), y_k^i, c_k] \right) \rightarrow \nabla_x O_{\bar{x}} \left(\sum_{i=1}^m \gamma[f_i(\bar{x})] \right) = 0$$

et donc la minimisation de $\tilde{G}(x)$ est asymptotiquement exacte. Nous pouvons donc conclure qu'asymptotiquement la procédure d'approximation pour le problème général est équivalente à la procédure d'approximation pour le problème (53), qui est de la forme de celui du paragraphe précédent, pour lequel on a établi les avantages de l'itération (23) pour ajuster les multiplicateurs.

On déduit donc que dans le cas général également, on a intérêt à employer l'itération proposée pour ajuster les multiplicateurs puisqu'elle possède les propriétés que nous connaissons bien.

Ceci termine le chapitre III ainsi que notre analyse de la procédure d'approximation proposée pour résoudre une classe particulière de problèmes d'optimisation non différentiables.

TROISIEME PARTIE :

L'IMPLEMENTATION DE L'ALGORITHME

DE LA METHODE DES MULTIPLICATEURS

Introduction

Dans cette troisième partie, nous décrirons comment nous avons implémenté l'algorithme de la méthode des multiplicateurs , en s'inspirant du programme de D.A. Pierre et M.J.Lowe (voir [12]).

Nous décrivons d'abord comment est construit le programme et le rôle des différentes sous-routines.

Ensuite suivront quelques renseignements utiles pour l'utilisateur, notamment l'entrée des données.

Et enfin, nous donnerons des résultats numériques.

CHAPITRE I

L'IMPLEMENTATION DE L'ALGORITHME1) L'algorithme général

Si nous nous rappelons l'algorithme de la méthode des multiplicateurs, nous constatons qu'il est une suite de cycles comportant chacun deux phases :

1. une phase de minimisation sans contrainte du lagrangien augmenté $L(., y_k, c_k)$ où y_k et c_k sont fixés
2. une phase d'ajustement des paramètres de pénalisation et des multiplicateurs y_k .

a) La phase de minimisation sans contrainte du lagrangien augmenté

* une direction de descente est d'abord générée
 -soit par la méthode DFP
 -soit par une méthode DFP- self-scaling
 selon l'option de l'utilisateur

On peut aussi faire une réinitialisation à la direction du gradient après chaque cycle.

* Lorsqu'on a une direction de descente, on recherche

le point qui minimise le lagrangien augmenté dans cette direction, à partir du point donné à l'étape précédente; autrement dit on fait une recherche unidirectionnelle. On obtient ainsi un nouveau point x .

On effectue alors plusieurs tests dans le but de déterminer si la phase de minimisation du lagrangien augmenté peut être considérée comme terminée (auquel cas on passe à la phase d'ajustement) ou non. Dans ce dernier cas, il faut recommencer le processus , c'est-à-dire chercher une nouvelle direction de descente, etc. On passe à la phase d'ajustement dans le cas où :

(1) le point trouvé n'apporte pas beaucoup de changement par rapport au point précédent (la phase de minimisation peut alors être considérée comme terminée) .C'est le cas lorsque :

$$\|x_{k-1} - x_k\| \leq \xi_3$$

et

$$\|\nabla L_{k-1} - \nabla L_k\| \leq EPV$$

$$\text{où } \nabla L_k = \nabla L(x_k, y_k, c_k)$$

ξ_3 est donné par l'utilisateur

$$10^{-2} \leq \xi_3 \leq 10^{-4}$$

$EPV = \xi_2$ au début du programme

$= \xi_1$ lorsqu'après une phase d'ajustement
on a eu $\|\nabla L\| \leq 10 \cdot \xi_2$

(2) ou bien on a atteint le nombre maximum

de recherches unidirectionnelles à l'intérieur d'un cycle : ce nombre est donné par l'utilisateur (en général $2n+1$).

(3) ou bien on a fait plus de n recherches unidirectionnelles dans le cycle et le gradient du lagrangien augmenté est suffisamment petit

$$\| \nabla L_k \| \leq \epsilon$$

(4) ou bien encore lorsque ^{une ou} plusieurs recherches unidirectionnelles ont été faites en ne produisant aucun point donnant une plus petite valeur du lagrangien augmenté.

b) La phase d'ajustement
.....

Les multiplicateurs sont ajustés par la formule :

$$y_{k+1} = y_k + c_k h(x_k)$$

dans le cas des contraintes d'égalités et

$$y_{k+1} = \max \{ 0, y_k + c_k g(x_k) \}$$

dans le cas des contrainte d'inégalité

Les paramètres de pénalisation sont ajustés par la formule

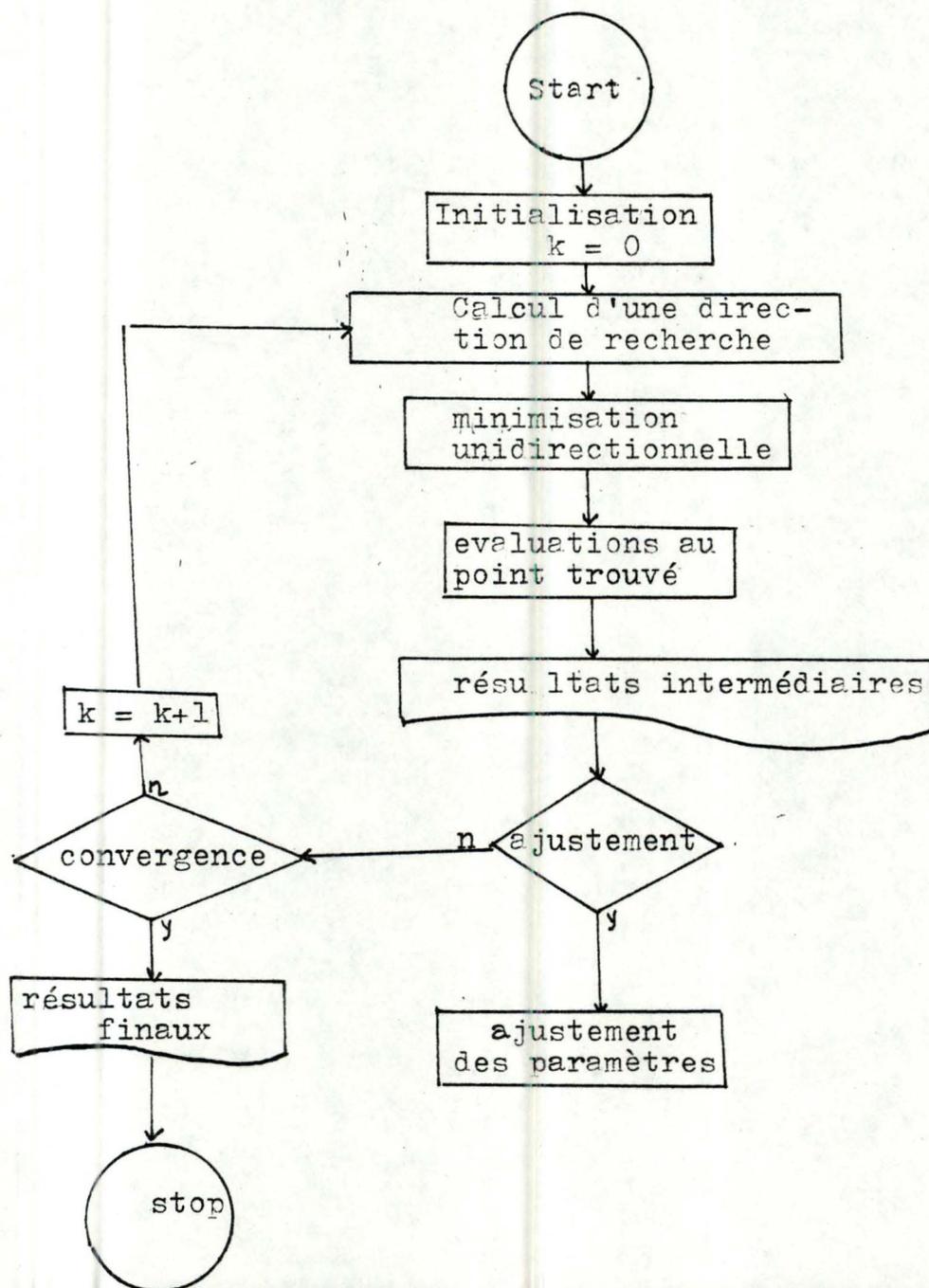
$$c_{k+1} = c_k \cdot w$$

où w est donné $w \geq 1$ (en général $w = 2$) ainsi que c_0 .

On effectue cette itération uniquement si c_{k+1} ne dépasse pas une limite donnée $wMAX$.

2) Construction du programme - Description sommaire des sous-routines

Le programme suivra donc l'organigramme suivant :



L'organigramme sera exécuté par la sous - routine LPNLP. Chacune des étapes de l'organigramme est exécutée par une ou plusieurs sous-routines.

LPNLP , suivant le schéma donné par l'organigramme, les appelle tour à tour jusqu'à la convergence ou l'arrêt du programme. Reprenons toutes ces étapes :

- i) Initialisation de l'algorithme
(effectuée par la sous-routine INITL)
- ii) Calcul de la direction de recherche
(sous-routine DFPRV)
- iii) Minimisation unidirectionnelle
(sous-routine SEARCH)
- iv) Evaluation du gradient du lagrangien augmenté au point x_{k+1}
(sous-routine GALAG)
Evaluation des fonctions objectives et des contraintes
(sous-routine FXNS, DELTA)
- v) Impression des résultats intermédiaires (en option)
(sous-routine OUTPUT)
- vi) Tests pour déterminer si la minimisation du lagrangien augmenté est terminée
- vii) Si oui , on fait un ajustement des multiplicateurs et des paramètres de pénalisation
(sous-routine UPDATE)

- viii) Si non et après la phase d'ajustement, on fait les tests de convergence
- ix) s'ils sont satisfait on imprime les résultats finaux (sous-routine OUTPUT) et le programme est terminé
- x) sinon on retourne à l'étape ii)

La sous-routine INITL

C'est cette sous-routine qui initialise l'algorithme. Elle initialise certains paramètres et lit les autres.

- (1) Tous les gradients et les multiplicateurs sont d'abord mis à zéro.
- (2) Lecture du point initial
des paramètres de pénalisation
- (3) Evaluation des fonctions du problème en ce point du lagrangien augmenté et de son gradient
- (4) Impression des conditions initiales et de toutes les variables importantes

La sous-routine CONGR

Elle est appelée par la sous-routine INITL et imprime les composantes constantes non nulles des gradients des fonctions objectives et des contraintes.

La sous-routine DFPRV

Elle calcule une direction de recherche par la

méthode de Davidon-Fletcher-Powell (voir [3] Davidon (1959) et [5] Fletcher et Powell (1963))

principales étapes de la sous-routine :

- (1) Tests pour déterminer si la direction de descente est engendrée par la méthode DFP ou est égale à l'opposé du gradient du lagrangien augmenté.

Ce dernier cas arrive :

- * si à l'étape précédente, on a pas trouvé de point x_k donnant une plus petite valeur du lagrangien augmenté
- * si on se trouve au début d'un nouveau cycle et qu'on a choisi une variante de la méthode consistant à réinitialiser la direction de descente à l'opposé du gradient après chaque cycle

- (2) Si la direction n'est pas celle de l'opposé du gradient, alors elle est engendrée par la méthode DFP ou une variante DFP-self-scaling (voir [10])

- (3) On vérifie si la direction engendrée est bien une direction de descente ($r_k \cdot \nabla L_k < 0$) sinon on pose $r_k = -\nabla L_k$ et on calcule la norme de cette direction.

La sous-routine SEARCH

Cette sous-routine calcule le pas t_k pour lequel $x_k + t_k r_k$ minimise $L(x_k + t_k r_k, y_k, c_k)$ en utilisant une procédure d'approximation quadratique.

La sous-routine VALUE

Elle est utilisée par la sous-routine SEARCH

dans la recherche unidirectionnelle.

D'abord elle évalue les fonctions du problème en appelant la sous-routine FXNS en un point-test $x_t = x_k + dr_k$ où d est une valeur du pas.

Ensuite, elle évalue le lagrangien augmenté en ce point, en appelant la sous-routine AULAG.

Cette valeur du lagrangien augmenté est alors comparée à la meilleure valeur obtenue dans la recherche jusqu'à présent et si elle est meilleure, on la retient à la place de l'autre.

La sous-routine DMIN

Elle calcule le point de minimum de l'approximation quadratique locale du lagrangien augmenté dans la direction de recherche.

La sous-routine DELTA

Elle est appelée par la sous-routine LPNLP lorsque la recherche unidirectionnelle est terminée. Elle calcule les différences

$$\Delta x_k = \|x_k - x_{k-1}\|$$

$$\Delta \nabla L_k = \|\nabla L_k - \nabla L_{k-1}\|$$

qui seront utilisées dans DFPRV à la prochaine étape.

Elle calcule aussi les normes de Δx_k , $\Delta \nabla L_k$ et ∇L_k puisqu'on en aura besoin dans les tests de convergence et pour déterminer s'il y a lieu d'ajuster c_k et y_k .

La sous-routine AULAG

Elle est appelée par INITL dans l'initialisation de l'algorithme

par SEARCH dans la recherche unidirectionnelle

par LPNLP après la phase d'ajustement

Elle évalue tout simplement le lagrangien augmenté en un point (x_k, y_k, c_k) .

La sous-routine GALAG

Elle est appelée également par les sous-routines INITL, SEARCH et LPNLP.

Elle évalue le gradient du lagrangien augmenté.

La sous-routine UPDATE

C'est la sous-routine d'ajustement.

Elle donne les valeurs de y_{k+1} et c_{k+1} suivant la procédure décrite plus haut.

La sous-routine OUTPUT

Elle imprime les résultats intermédiaires après k recherches unidirectionnelles (où k est choisi par l'utilisateur) et à la fin du programme , elle imprime les résultats finaux.

CHAPITRE II

L'UTILISATION DU PROGRAMME

Nous allons expliquer dans ce chapitre comment l'utilisateur doit procéder s'il veut résoudre un problème de programmation non linéaire avec contraintes à l'aide du programme.

Le programme a été conçu afin de faciliter au maximum l'utilisation du programme. En effet, pour passer d'un programme à l'autre, on ne devra modifier que quelques cartes dans le programme principal pour donner les dimensions exactes des vecteurs spécifiques au problèmes et donner l'expression analytique des fonctions objective et des contraintes ainsi que leurs gradients dans deux sous-routines ; les autres paramètres sont entrés dans les données. Toutes les autres sous-routines décrites précédemment n'ont besoin d'aucune modification.

Voyons cela un peu plus en détail.

1) Le programme principal

Il est très court et comportera les cartes suivantes:

```

1      IMPLICIT REAL*8(A-H,O-Z)
2      DIMENSION DLX(N),DLG(N),GAL(N),GF(N),GO(N),
3      1 IX(N),R(N),X(N),XT(N),H(N*N)
4      DIMENSION HI(NE),HIS(NE),EMULT(NE),GE(N*NE)
5      DIMENSION GI(NI),GIS(NI),OMULT(NI),GIN(N*NI)

```

```

6      DIMENSION C(N),D(N),XMULTL(N),XMULTU(N)
7      CALL LPNLP(C,D,DLX,DLG,HI,HIS,GI,GIS,GAL,GE,GF,
8      1 GIN,GO,H,EMULT,OMULT,XMULTL,XMULTU,IX,R,X,XT)
9      STOP
10     END

```

Il n'y a que les cartes comportant l'instruction "DIMENSION" qui doivent être changées : l'utilisateur remplacera les lettres N,NE,NI,N*N,N*NE,N*NI par les valeurs effectives propres au problème ;

N = nombre de variables x_i

NE = nombre de contraintes d'égalité

NI = nombre de contraintes d'inégalité

Remarque

Les contraintes du genre :

$c_i \leq x_i \leq d_i$ ou $c_i \leq x_i$ ou $x_i \leq d_i$ ne sont pas considérées comme des contraintes d'inégalité en ce sens qu'elles sont traitées à part dans le programme.

NI ne comptera donc pas le nombre de contraintes de ce type.

Si NE = 0 , on aura pas besoin de mettre la carte 4

Si NI = 0 , on aura pas besoin de mettre la carte 5

S'il n'y a pas de contrainte du genre variable bornée , on a pas besoin de mettre la carte 6.

En procédant de cette manière , il ne faudra faire aucune modification dans les autres sous-routines puisqu'on peut utiliser l'instruction :

```
DIMENSION VECT(1)
```

2) La sous-routine FXNS

On y exprime les fonctions du problème.
Elle comporte les cartés suivantes:

```

SUBROUTINE FXNS (X,F,HI,GI)
DIMENSION X(1),HI(1),GI(1)
F=
HI(1)=
  ⋮
HI(NE)=
GI(1)=
  ⋮
GI(NI)=
RETURN
END

```

On exprime d'abord la fonction objective (F), puis les NE contraintes d'égalité (si NE=0 , on ne met rien) et enfin les NI contraintes d'inégalité (si NI=0 on ne met rien).

Remarques

- 1) Rappelons que les variables bornées ne sont pas considérées comme des contraintes d'inégalité : elles sont traitées dans le programme par l'intermédiaire des données.
- 2) On considère des contraintes de la forme :

$$h_i(x) = 0 \quad \text{pour les contraintes d'égalité et}$$

$$g_i(x) \leq 0 \quad \text{pour les contraintes d'inégalité.}$$

Donc si on a des contraintes de la forme :

$h_i(x) = a_i$ pour les contraintes d'égalité ou
 $g_i(x) \leq b_i$ pour les contraintes d'inégalité ,
 il ne faudra pas oublier de faire passer la constante
 a_i ou b_i dans le membre de gauche quand on exprimera
 les fonctions dans le programme. De même on veillera
 à ce que pour les contraintes d'inégalité , le signe d'iné-
 galité soit tourné dans le même sens que ci-dessus.

3) La sous-routine GRAD

On y exprime les composantes non constantes des
 gradients des fonctions du problème. Elle comporte les
 cartes suivantes :

```

SUBROUTINE GRAD(X,GF,GE,GIN)
DIMENSION X(1),GF(1),GE(1),GIN(1)
GF(1)=
GF(2)=
⋮
GF(N)=
GE(1)=
⋮
GE(N≠NE)=
GIN(1)=
⋮
GIN(N≠NI)
RETURN
END

```

On exprime d'abord les n composantes du gra-
 dient de F (GF). Si l'une (ou plusieurs ou même toutes)

est constante , on ne la met pas.

Ensuite , de la même manière on exprime les composantes non constantes du gradient des contraintes d'égalité en commençant par les N composantes du gradient de la première contrainte d'égalité, puis suivent les N composantes du gradient de la deuxième contrainte ,etc.

Enfin , on traite de la même manière les composantes non constantes du gradient des contraintes d'inégalité.

Remarquons que pour un problème linéaire , cette sous-routine est vide.

4) Les données

Nous décrivons dans l'ordre quelles variables vont figurer dans les données et suivant quel format il faut les écrire.

(a) Sur la première carte, l'utilisateur écrira un commentaire : le titre du problème ou son numéro ou toute autre information comme l'auteur, la description du problème, s'il est linéaire ou non, avec contraintes, etc.

Attention ! on ne dispose que d'une seule carte pour ces commentaires mais cette carte doit obligatoirement figurer en tête des données (on peut la remplacer par une carte blanche).

(b) Sur la deuxième carte on indiquera suivant le FORMAT (10I5) les variables suivantes :

N, NE, NI, NBV, IMAX, NPRINT, NUP, NSRCH, ISS, IRESET

- N = nombre de variables x_i
- NE = nombre de contraintes d'égalité
- NI = nombre de contraintes d'inégalité
- NBV = nombre de variables bornées (soit inférieurement, soit supérieurement, ou soit des 2 manières)
NBV est au maximum égal à N.
- IMAX = nombre maximum de recherches unidirectionnelles permises par l'utilisateur : le programme s'arrête lorsque IMAX recherches unidirectionnelles ont été faites.
- NPRINT : Le programme imprime automatiquement les informations disponibles au début du programme et les résultats finaux. Si l'utilisateur ne désire rien d'autre il pose NPRINT = 0. S'il veut l'impression des résultats toutes les J recherches , il pose NPRINT = J où $J \in \{1, 2, 3, \dots, IMAX\}$
- NUP = si NUP=0 , le programme imprime les informations après chaque phase d'ajustement des multiplicateurs et des paramètres de pénalisation.
si NUP = 1 , le programme n'imprime pas ces informations.
- NSRCH = nombre maximum de recherches unidirectionnelles pour un cycle (d'habitude on pose NSRCH = 2N+1)
- ISS = 0 : l'utilisateur choisit la méthode DFP comme méthode de recherche d'une direction de descente
= 1 : il choisit la méthode DFP-SS.
- IRESET = 1 : l'utilisateur choisit de réinitialiser la direction de descente à l'opposé du gradient à la fin de chaque cycle à condition qu'au moins

N recherches unidirectionnelles aient été faites depuis la dernière réinitialisation.

IRESET = 0 : l'utilisateur choisit de ne pas réinitialiser.

(c) On indiquera suivant le FORMAT (7I5) les variables :

IGF,IGE,IGI,IEMULT,IOMULT,IXMLTL,IXMLTU

IGF = nombre de composantes constantes non nulles du gradient de la fonction objective.

IGE = nombre de composantes constantes non nulles des gradients des fonctions exprimant les contraintes d'égalité

IGI = idem pour les contraintes d'inégalité

IEMULT = nombre de multiplicateurs des contraintes d'égalité qui ne sont pas initialisés à 0 (= 0 si NE = 0 ou s'ils sont tous initialisés à zéro , ce que l'on fait généralement)

IOMULT = idem pour les contraintes d'inégalité

IXMLTL = idem pour les multiplicateurs des variables bornées inférieurement

IXMLTU = idem pour les variables bornées supérieurement.

(d) On indiquera suivant le FORMAT(3D10.5) les variables

EPS1, EPS2, EPS3

EPS1 = partie du critère de convergence
on a la convergence lorsque :

$$\|\nabla L_k\| \leq \text{EPS1} \quad \text{et} \quad \|x_{k-1} - x_k\| \leq \text{EPS3}$$

En général $10^{-8} \leq \text{EPS1} \leq 10^{-4}$

EPS2, EPS3 = partie du critère d'ajustement
on fait un ajustement quand :

$$\|x_k - x_{k-1}\| \leq \text{EPS3}$$

et

$$\|\nabla L_k\| \leq \text{EPV}$$

où EPV = EPS2 au début du programme
puis est réduit dynamiquement à EPS1.
EPS2 peut être différent de EPS3 et
en général :

$$10^{-4} \leq \text{EPS2} \leq 10^{-2}$$

$$10^{-4} \leq \text{EPS3} \leq 10^{-2}$$

- (e) Sur les cartes suivantes on indique les composantes du point de départ X suivant le FORMAT(8D10.5) ; on met donc 8 composantes par carte et on utilise autant de cartes qu'il le faut.
- (f) Si NE+NI+NBV=0, c'est-à-dire si le problème est sans contraintes aller en (g).
Sinon suivant le FORMAT(5D10.5) indiquer les variables :
W1, W2, W1MAX, W2MAX, WF
- W1 = paramètre de pénalisation initial pour les contraintes d'égalité
W2 = paramètre de pénalisation initial pour les contraintes d'inégalité
W1MAX = valeur maximale que peut prendre le paramètre de pénalisation pour les contraintes d'égalité.

W2MAX = valeur maximale que peut prendre le paramètre de pénalisation pour les contraintes d'inégalité.

WF = facteur d'accroissement du paramètre de pénalisation

(pour avoir une idée des valeurs à donner à ces paramètres voir le chapitre suivant)

(g) Si IGF = 0 aller en (h).

Sinon , suivant le FORMAT(4(I5,D15.5)) , indiquer la valeur des composantes constantes non nulles du gradient de F en indiquant chaque fois avant cette valeur le numéro de la composante. On peut donc mettre 4 données par carte.

(h) Si IEMULT = 0 aller en (i).

Sinon , suivant le FORMAT(4(I5,D15.5)) , indiquer la valeur des multiplicateurs initiaux non nuls en écrivant chaque fois avant cette valeur le numéro de la contrainte d'égalité correspondante.

(i) Si IGE = 0 aller en (j).

Sinon, suivant le FORMAT (4(I5,D15.5)) , indiquer en procédant de la même manière que décrit en (g) les composantes constantes non nulles des gradients des contraintes d'égalité.

(j) Si IOMULT = 0 aller en (k)

Sinon , en procédant exactement de la même manière que décrit en (h) , indiquer la valeur des multiplicateurs non nuls correspondants aux contraintes d'inégalité.

(k) Si $IGI = 0$ aller en (l)

Sinon , en procédant exactement de la même manière que décrit en (g) , indiquer les composantes constantes non nulles des gradients des fonctions exprimant les contraintes d'inégalité.

(l) Si $NBV = 0$, on a terminé d'entrer les données.

Sinon, suivant le `FORMAT(2(I5,I5,D15.5))` , on va exprimer les contraintes du type variable bornée en procédant de la manière suivante :

- on indique d'abord le numéro de la composante:K
- ensuite on indique le code correspondant $IX(K)$
 - $IX(K) = 0$: la variable $X(K)$ n'est pas bornée
 - $IX(K) = 1$: $X(K)$ est bornée inférieurement uniquement
 - $IX(K) = 2$: $X(K)$ est bornée supérieurement uniquement
 - $IX(K) = 3$: $X(K)$ est bornée inférieurement et supérieurement
- puis on indique la borne inférieure $C(K)$ - s'il n'y en a pas on laisse 15 blancs - et enfin la borne supérieure $D(K)$. Remarquons que seules les variables n'ayant pas le code $IX(K)=0$ sont ainsi représentées dans les données.

(m) Si $IXMLTL = 0$ aller en (n).

Sinon , en procédant exactement de la même manière que décrit en (h) , indiquer les multiplicateurs initiaux non nuls correspondant aux variables bornées inférieurement.

(n) Si $IXMLTU = 0$, on a terminé d'entrer les données.
Sinon , en procédant exactement de la même manière
que décrit en (h) , indiquer les multiplicateurs ini-
tiaux non nuls correspondants aux variables bornées
supérieurement.

CHAPITRE III

RESULTATS NUMERIQUES

Nous avons testé le programme sur 4 problèmes de petite taille (le code ne donne pas de résultats satisfaisants pour un plus grand nombre de variables et de contraintes) avec des conditions initiales différentes (voir tableaux et résultats commentés plus loin).

Problème A

$$\text{minimiser } -x_1 \cdot x_2$$

$$x_1 + x_2 \geq 0$$

$$x_1 + x_2^2 \leq 1$$

$$\text{Point optimal : } x_1^* = \frac{2}{3} \cong 0.666667$$

$$x_2^* = \frac{1}{\sqrt{3}} \cong 0.57735$$

pour lequel la valeur de la fonction objective est

$$F^* = -0.3849$$

les multiplicateurs optimaux valent :

$$y_1^* = 0.$$

$$y_2^* = \frac{1}{\sqrt{3}} \cong 0.57735$$

Conditions initiales communes à tous les tests effectués sur le problème A :

- point de départ : $x_1 = 1.$, $x_2 = 1.$
- multiplicateurs de départ : $y_1 = y_2 = 0.$
- $EPS1 = 10^{-6}$, $EPS2 = EPS3 = 10^{-2}$
- nombre maximum de recherches unidirectionnelles permises dans un cycle = NSRCH = 5

Autres conditions initiales portant

- a) sur la méthode de recherche d'une direction de

 descente

les paramètres de pénalisation sont les mêmes pour les quatre tests , à savoir :

- paramètre de pénalisation de départ : $C_0 = 2.$
- facteur d'accroissement du paramètre de pénalisation : $WF = 2.$ ($c_{k+1} = c_k \cdot WF$)
- limite du paramètre de pénalisation : $C_{MAX} = 100.$

On a testé les quatre méthodes suivantes :

- A1 : DFP-SS avec réinitialisation à la direction de l'opposé du gradient après chaque cycle
- A2 : DFP-SS sans réinitialisation
- A3 : DFP avec réinitialisation
- A4 : DFP sans réinitialisation

- b) sur différentes manières d'ajuster les paramètres

 de pénalisation

pour les quatre tests , la méthode de recherche d'une direction de descente est la même , à savoir :
 DFP-SS avec réinitialisation

On a fait les quatre tests suivants :

A1 : $C_0 = 2.$, $WF = 2.$, $C_{MAX} = 100.$

A5 : $C_0 = 2.$, $WF = 2.$, $C_{MAX} = 16.$

A6 : $C_0 = 3.$, $WF = 3.$, $C_{MAX} = 81.$

A7 : $C_0 = 3.$, $WF = 2.$, $C_{MAX} = 24.$

Problème B (proposé par D.A.Pierre)

minimiser $-x_2$

$$x_1^2 + x_2^2 + x_3^2 = 1.$$

$$2x_2 - x_1 \leq 1.$$

A l'optimum : $x_1^* = \frac{3}{5}$, $x_2^* = \frac{4}{5}$, $x_3^* = 0.$

$$F^* = -\frac{4}{5}$$

$$y_1^* = \frac{1}{4} , y_2^* = \frac{3}{10}$$

Conditions initiales communes

- point de départ : $x_1 = -0.1$, $x_2 = -1.$, $x_3 = 0.1$
- multiplicateurs de départ : $y_1 = y_2 = 0.$
- $EPS1 = 10^{-6}$, $EPS2 = EPS3 = 10^{-2}$
- NSRCH = 7

Autres conditions initiales portant

a) sur la méthode de recherche d'une direction de descente

avec $C_0 = 0.25$, $WF = 2.$, $C_{MAX} = 10^5$

- B1 : DFP-SS avec réinitialisation
 B2 : DFP-SS sans réinitialisation
 B3 : DFP avec réinitialisation
 B4 : DFP sans réinitialisation

b) sur différentes manières d'ajuster les
paramètres de pénalisation

avec DFP sans réinitialisation

- B1 : $C_0 = 0.25$, $WF = 2.$, $C_{MAX} = 10^5$
 B5 : $C_0 = 0.5$, $WF = 2.$, $C_{MAX} = 10^5$
 B6 : $C_0 = 1.$, $WF = 2.$, $C_{MAX} = 10^5$
 B7 : $C_0 = 2.$, $WF = 2.$, $C_{MAX} = 10^5$

Problème C (Beale)

$$\min -8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3 + 9$$

$$x_1 + x_2 + 2x_3 \leq 3$$

$$x_1 \geq 0 , x_2 \geq 0 , x_3 \geq 0$$

A l'optimum : $x_1^* = \frac{4}{3}$, $x_2^* = \frac{7}{9}$, $x_3^* = \frac{4}{9}$

$$F^* = -\frac{1}{9}$$

$$y_1^* = \frac{2}{9} , y_2^* = y_3^* = y_4^* = 0$$

Conditions initiales communes

- point de départ : $x_1 = x_2 = x_3 = 0.5$

- multiplicateurs initiaux: $y_1 = y_2 = y_3 = y_4 = 0$
- $EPS1 = 10^{-6}$, $EPS2 = EPS3 = 10^{-2}$
- $NSRCH = 7$

Autres conditions initiales portant :

a) sur la méthode de recherche d'une direction de

 descente

avec $C_0 = 1$, $WF = 2$, $C_{MAX} = 10^4$

- C1 : DFP-SS avec réinitialisation
- C2 : DFP-SS sans réinitialisation
- C3 : DFP avec réinitialisation
- C4 : DFP sans réinitialisation

b) sur différentes manières d'ajuster les para-

 mètres de pénalisation

avec la méthode DFP avec réinitialisation

- C3 : $C_0 = 1$, $WF = 2$, $C_{MAX} = 10^4$
- C5 : $C_0 = 1$, $WF = 4$, $C_{MAX} = 10^4$
- C6 : $C_0 = 5$, $WF = 2$, $C_{MAX} = 10^4$
- C7 : $C_0 = 10$, $WF = 2$, $C_{MAX} = 10^4$
- C8 : $C_0 = 20$, $WF = 2$, $C_{MAX} = 10^4$
- C9 : $C_0 = 50$, $WF = 2$, $C_{MAX} = 10^4$
- C10 : $C_0 = 100$, $WF = 2$, $C_{MAX} = 10^4$
- C11 : $C_0 = 1$, $WF = 2$, $C_{MAX} = 10^4$
- C12 : $C_0 = 1$, $WF = 2$, $C_{MAX} = 10^4$
- C13 : $C_0 = 1$, $WF = 3$, $C_{MAX} = 10^4$

Problème D (Fiacco et Mc Cormick)

$$\min \frac{1}{3} (x_1 + 1)^3 + x_2$$

$$x_1 \geq 1, x_2 \geq 0$$

A l'optimum : $x_1^* = 1, x_2^* = 0$

$$F^* = \frac{8}{3}$$

$$y_1^* = 4, y_2^* = 1$$

Conditions initiales communes

- point de départ : $x_1 = 1.125, x_2 = 0.125$
- multiplicateurs de départ : $y_1 = y_2 = 0$
- EPS3 = EPS2 = 10^{-2} , EPS1 = 10^{-6}
- NSRCH = 5

Autres conditions initiales portant :

- a) sur la méthode de recherche d'une direction de descente

Avec $C_0 = 1, WF = 2, CMAX = 10^4$

- D1 : DFP-SS avec réinitialisation
- D2 : DFP-SS sans réinitialisation
- D3 : DFP avec réinitialisation
- D4 : DFP sans réinitialisation

- b) sur différentes manières d'ajuster les paramètres de pénalisation

avec la méthode DFP sans réinitialisation

D4 : $C_0 = 1$, WF = 2 , CMAX = 10^4
 D5 : $C_0 = 1$, WF = 4 , CMAX = 10^4
 D6 : $C_0 = 5$, WF = 2 , CMAX = 10^4
 D7 : $C_0 = 3$, WF = 2 , CMAX = 10^4

LES RESULTATS DE TOUS LES TESTS DECRITS CI-DESSUS
 SE TROUVENT A LA FIN DU CHAPITRE.

Etude des résultats numériques

1) la méthode de recherche d'une direction de descente

Problème	nb. cycles	nb.rech. unidir.	nb.appels FXNS	nb.appels GRAD
A1	9	30	109	31
A2	9	30	109	31
A3	7	22	70	23
A4	7	22	70	23
B1	15	100(*)	398	101
B2	15	100(*)	398	101
B3	11	31	89	32
B4	11	31	89	32
(*) : limite du nb. de recherches unidirectionnelles				
C1	10	56	189	57
C2	10	56	189	57
C3	7	28	88	29
C4	7	28	88	29
D1	11	27	90	28
D2	11	27	90	28
D3	11	25	76	26
D4	11	25	76	26

On constate une similitude importante entre tous les tableaux :

- 1) du moins pour les problèmes que nous avons testés faire une réinitialisation après chaque cycle ou ne pas en faire revient au même
- 2) La méthode DFP donne de meilleurs résultats que DFP-SS.

Pour le problème B , on n'a même pas atteint la convergence en 100 itérations avec DFP-SS. Pour le problème C , il a fallu le double de recherches unidirectionnelles et le double d'évaluations des fonctions.

2) La manière d'ajuster les paramètres de pénalisation

Problème A

Pb.	C ₀	WF	C _{MAX}	nb. cycles	nb rech. unid.	nb. appels FXNS	nb. appels GRAD
A1	2	2	100	9	30	109	31
A5	2	2	16	12	45	158	46
A6	3	3	81	12	50	169	51
A7	3	2	24	20	100*	361	101

* on a atteint le nombre maximum de recherches unidirectionnelles sans avoir la convergence

Interprétation

- En comparant A1 et A5 , on constate qu'on a choisi la même manière d'ajuster les paramètres de pénalisa-

tion sauf que la limite maximum du paramètre pour A5 est plus petite que celle de A1. On constate que la convergence est plus rapide pour A1 que pour A5 : c'est bien ce qu'on attendait puisque la convergence est linéaire avec un coefficient de convergence inversement proportionnel à CMAX.

- La meilleure manière d'ajuster les paramètres parmi les 4 proposées est la première. Ce n'est pas surprenant puisque c'est celle qui a le plus grand CMAX.

Problème B

Pb.	C ₀	WF	CMAX	nb. cycles	nb. rech. unid.	nb. appels FXNS	nb. appels GRAD
B1	0.25	2.	10 ⁵	15	100*	398	101
B5	0.5	2.	10 ⁵	6	25	65	26
B6	1.	2.	10 ⁵	8	35	111	35
B7	2.	2.	10 ⁵	9	47	157	48

* on a atteint le nombre maximum de recherches unidirectionnelles sans avoir la convergence.

Ici on a laissé croître le paramètre de pénalisation puisqu'il n'a jamais atteint la limite maximum (voir les résultats plus loin). On a concentré l'attention sur les valeurs de départ. La théorie nous dit que le paramètre de pénalisation doit être assez grand pour satisfaire les théorèmes et cependant il ne doit pas être trop grand pour ne pas produire un problème mal conditionné. Dans le cas présent, si WF = 2. , la meilleure valeur de départ est 0.5 .

Pb.	C ₀	WF	C _{MAX}	nb. cycles	nb. rech. unid.	nb. appels FXNS	nb. appels GRAD
C3	1.	2.	10 ⁴	7	28	88	29
C5	1.	4.	10 ⁴	11	41*	180	41
C6	5.	2.	10 ⁴	5	18	52	19
C7	10.	2.	10 ⁴	11	31*	178	28
C8	20.	2.	10 ⁴	21	85	407	84
C9	50.	2.	10 ⁴	19	80	349	81
C10	100.	2.	10 ⁴	19	61	337	61
C11	1.	2.	50.	7	28	84	29
C12	1.	2.	100.	7	28	88	29
C13	1.	3.	10 ⁴	5	16	43	17

* le programme a été arrêté avant d'avoir la convergence (4 fois de suite, la recherche unidirectionnelle n'a produit aucun point donnant une plus petite valeur du lagrangien augmenté).

1) Point de vue des valeurs de départ (WF = 2)

Comparons C3, C6, C7, C8, C9, C10

C'est pour C₀=5 que le test produit le meilleur résultat. Pour C₀ plus petit (C₀ = 1) les résultats sont bons aussi. Pour C₀ plus grand, le problème devient plus mal conditionné et la convergence, si elle est atteinte, est plus lente.

2) Point de vue facteur d'accroissement

Comparons C3, C13, C5

C'est WF=3 qui donne le meilleur résultat. Pour WF plus petit (WF = 2) le résultat est bon aussi mais pour WF=4, le paramètre de pénalisation grandit trop vite et produit un problème mal conditionné.

3) Point de vue de la limite supérieure du paramètre de pénalisation

Comparons C3, C11 et C12.

On obtient à peu près les mêmes résultats. Ceci est dû au fait que lorsqu'on laisse croître le paramètre de pénalisation de cette manière, et pour ce problème particulier, la convergence est atteinte assez rapidement et donc le paramètre de pénalisation n'a pas le temps de prendre des valeurs très grandes (CMAX = 128)

Problème D

Pb.	C ₀	WF	CMAX	nb. cycles	nb. rech. unid.	nb. appels FXNS	nb. appels GRAD
D4	1.	2.	10 ⁴	11	25	76	26
D5	1.	4.	10 ⁴	8	20	69	21
D6	5.	2.	10 ⁴	16	27	115	27
D7	3.	2.	10 ⁴	9	21	71	22

- Comparant D4, D5, on voit qu'en multipliant par 4, on obtient un meilleur résultat qu'en multipliant par 2.

- Comparant D4, D7 et D6, on voit que c'est pour C₀ = 3 qu'on obtient le meilleur résultat. Pour C₀ = 1., le résultat est bon aussi mais pour C₀ = 5., la convergence est nettement moins rapide.

Conclusion

Il n'y a pas de manière "optimale" d'ajuster les paramètres de pénalisation qui soit valable pour tous les problèmes; mais l'analyse des résultats semble indiquer

que :

1) le paramètre de pénalisation de départ doit être choisi assez petit mais encore une fois cela dépend du problème. On peut cependant remarquer , d'après nos observations qu'il vaut mieux le choisir trop petit plutôt que trop grand. De plus , les tests nous donnent une idée de l'intervalle où il sera choisi.

2) le facteur d'accroissement ne doit pas être trop grand non plus. Il semble qu'une bonne valeur soit $WF = 2$. Mais il se peut que dans certains cas une plus grande valeur (3 ou 4) de WF donne des résultats un peu meilleurs. Comme pour le choix de C_0 , il vaut mieux le choisir trop petit plutôt que trop grand.

Remarque très importante

Ces conclusions ont été tirées en observant les résultats de quelques petits problèmes. Mais on ne sait dans quelle mesure ils sont valables pour les autres problèmes et ne présentent donc aucun caractère d'universalité ou de véracité. Il faudrait encore beaucoup d'observations pour pouvoir en faire des règles générales. Et de toutes façons , le programme ne donne des résultats satisfaisants que pour les problèmes de petite taille.

Dans les pages qui suivent , on trouvera les résultats finaux complets de tous les problèmes que nous avons testés.

Pour chaque test , nous avons rappelé les conditions initiales non communes à tous les tests effectués sur un même problème. Nous avons aussi ajouté le nombre de cycles complets effectués par le programme pour chaque problème-test.

Lecture des résultatsTable des variables

SEARCH POINT : nombre total de recherches unidirectionnelles

FXNS CALLS : nombre total d'évaluations des fonctions
du problème

GRAD CALLS : nombre total d'évaluations des gradients
des fonctions du problème

F = valeur finale de la fonction objective

AUG.LAG. : valeur finale du lagrangien augmenté L_k

GMAG : norme du gradient du lagrangien augmenté calculé
au dernier point $\|\nabla L_k\|$

SEARCH DIRECTION TO THIS POINT : composantes de la der-
nière direction de recherche r_k

RHO : valeur du pas : $x_k = x_{k-1} + \text{RHO } r_k$

DELX : norme de la différence entre les deux derniers
points $\|x_k - x_{k-1}\|$

DELG : norme de la différence entre les gradients du
lagrangien augmenté évalués en les deux derniers
points $\|\nabla L_{k-1} - \nabla L_k\|$

X VALUES : valeur du point final

EQUALITY CONSTRAINT STATUS

J : numéro de la contrainte

GI(J) : valeur de la contrainte au dernier point

OMULT(J) : valeur du multiplicateur à la dernière itération

INEQUALITY CONSTRAINT STATUS : idem

CONSTRAINT VARIABLE STATUS

K : numéro de la variable

C(K)-X(K) : différence entre la composante et sa borne inférieure

XMULTL(K) : multiplicateur associé à cette contrainte

X(K)-D(K) : différence entre la composante et sa borne supérieure

XMULTU(K) : multiplicateur associé à cette contrainte

W1 : valeur finale du paramètre de pénalisation pour les contraintes d'égalité

W2 : valeur finale du paramètre de pénalisation pour les contraintes d'inégalité.

..... FINAL SEARCH POINT VALUES TO FOLLOW

Résultats du problème A1

***** SEARCH POINT *** 30 *****

FXNS CALLS... 109 GRAD CALLS... 31 RHO = 0.100000D-02

F = -0.384900D 00 AUG. LAG. = -0.384900D 00 GMAG = 0.912971D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
 -0.106148D-05 -0.305452D-07

DELX = 0.106192D-08 DELG = 0.166737D-06 RMAG = 0.106192D-05

X VALUES, X(1), ..., X(N)
 0.666667D 00 0.577350D 00

INEQUALITY CONSTRAINT STATUS :

J	GI(J)	OMULT(J)
1	-0.521746D-09	0.577351D 00
2	-0.124402D 01	0.000000D 00

W1 = 0.000000D 00 W2 = 0.100000D 03

* méthode de recherche d'une direction de descente :

DFF-SS avec réinitialisation

* paramètre de pénalisation de départ : $C_0 = 2$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 100.$

Nombre de cycles : 9

----- FINAL SEARCH POINT VALUES TO FOLLOW -----

Résultats du problème A2

***** SEARCH POINT *** 30 *****

FXNS CALLS... 109 GRAD CALLS... 31 RHO = 0.100000D-02

F = -0.384900D 00 AUG. LAG. = -0.384900D 00 GMAG = 0.912971D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
 -0.106148D-05 -0.305452D-07

DELX = 0.106192D-08 DELG = 0.166737D-06 RMAG = 0.106192D-05

X VALUES, X(1), ..., X(N)
 0.666667D 00 0.577350D 00

INEQUALITY CONSTRAINT STATUS :

J	G1(J)	OMULT(J)
1	-0.521746D-09	0.577351D 00
2	-0.124402D 01	0.000000D 00

W1 = 0.000000D 00 W2 = 0.100000D 03

* méthode de recherche d'une direction de descente :

DFP-SS sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 2$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 100$

Nombre de cycles : 9

..... FINAL SEARCH POINT VALUES TO FOLLOW

Résultats du problème A3

***** SEARCH POINT *** 22 *****

FXNS CALLS... 70 GRAD CALLS... 23 RHO = 0.209579D-02

F = -0.384900D 00 AUG. LAG. = -0.384900D 00 GMAG = 0.781346D-06

SEARCH DIRECTION TO THIS POINT,R(1),...,R(N) NG = 1
 -0.102002D-05 -0.116055D-05

DELX = 0.323820D-08 DELG = 0.754478D-06 RMAG = 0.154510D-05

X VALUES,X(1),...,X(N)
 0.666667D 00 0.577350D 00

INEQUALITY CONSTRAINT STATUS :

J	GI(J)	OMULT(J)
1	-0.612297D-10	0.577351D 00
2	-0.124402D 01	0.000000D 00

W1 = 0.000000D 00 W2 = 0.100000D 03

* méthode de recherche d'une direction de descente :

DFP avec réinitialisation

* paramètre de pénalisation de départ : $C_0 = 2$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : 100

Nombre de cycles : 7

----- FINAL SEARCH POINT VALUES TO FOLLOW -----

Résultats du problème A4

***** SEARCH POINT ** 22 *****

FXNS CALLS... 70 GRAD CALLS... 23 RHO = 0.209579D-02

F = -0.384900D 00 AUG. LAG. = -0.384900D 00 GMAG = 0.781346D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
 -0.102002D-05 -0.116055D-05

DELX = 0.323820D-08 DELG = 0.754478D-06 RMAG = 0.154510D-05

X VALUES, X(1), ..., X(N)
 0.666667D 00 0.577350D 00

INEQUALITY CONSTRAINT STATUS :

J	GI(J)	OMULT(J)
1	-0.612297D-10	0.577351D 00
2	-0.124402D 01	0.000000D 00

W1 = 0.000000D 00 W2 = 0.100000D 03

* méthode de recherche d'une direction de descente :

DFP sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 2$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 100$

Nombre de cycles : 7

..... FINAL SEARCH POINT VALUES TO FOLLOW

Résultats du problème A5

*****SEARCH POINT *** 45*****

FXNS CALLS... 158 GRAD CALLS... 46 RHO = 0.347853D-01

F = -0.384900D 00 AUG. LAG. = -0.384900D 00 GMAG = 0.996045D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
 -0.103408D-05 -0.405512D-06

DELX = 0.386378D-07 DELG = 0.125312D-05 RMAG = 0.111075D-05

X VALUES, X(1), ..., X(N)
 0.666667D 00 0.577350D 00

INEQUALITY CONSTRAINT STATUS :

J	GI(J)	OMULT(J)
1	-0.238771D-07	0.577350D 00
2	-0.124402D 01	0.000000D 00

W1 = 0.000000D 00 W2 = 0.160000D 02

* méthode de recherche d'une direction de descente :

DFP-SS avec réinitialisation

* paramètre de pénalisation de départ : $C_0 = 2$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 16$

Nombre de cycles : 12

..... FINAL SEARCH POINT VALUES TO FOLLOW

Résultats du problème A6

***** SEARCH POINT *** 50 *****

FXNS CALLS... 169 GRAD CALLS... 51 RHO = 0.530690D-02

F = -0.384900D 00 AUG. LAG. = -0.384900D 00 GMAG = 0.538215D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
0.258230D-04 0.308620D-04

DELX = 0.213552D-06 DELG = 0.402879D-04 RMAG = 0.402404D-04

X VALUES, X(1), ..., X(N)
0.666667D 00 0.577350D 00

INEQUALITY CONSTRAINT STATUS :

J	G1(J)	OMULT(J)
1	0.504805D-09	0.577350D 00
2	-0.124402D 01	0.000000D 00

W1 = 0.000000D 00 W2 = 0.810000D 02

* méthode de recherche d'une direction de descente :

DFP-SS avec réinitialisation

* paramètre de pénalisation de départ : $C_0 = 3$

facteur d'accroissement : $WF = 3$

paramètre de pénalisation limite : $C_{MAX} = 81$

Nombre de cycles : 12

----- FINAL SEARCH POINT VALUES TO FOLLOW -----

Résultats du problème A7

***** SEARCH POINT *** 100 *****

FXNS CALLS... 361 GRAD CALLS... 101 RHO = 0.218914D-01

F = -0.384900D 00 AUG. LAG. = -0.384900D 00 GMAG = 0.134818D-04

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
 -0.772599D-05 -0.286116D-04

DELX = 0.648783D-06 DELG = 0.327351D-04 RMAG = 0.296364D-04

X VALUES, X(1), ..., X(N)
 0.666660D 00 0.577355D 00

INEQUALITY CONSTRAINT STATUS :

J	GI(J)	OMULT(J)
1	-0.382566D-06	0.577352D 00
2	-0.124402D 01	0.000000D 00

W1 = 0.000000D 00 W2 = 0.240000D 02

- * méthode de recherche d'une direction de descente :
 DFP-SS avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 3$
 facteur d'accroissement : $WF = 2$
 paramètre de pénalisation limite : $C_{MAX} = 24$

Nombre de cycles : 20

Résultats du problème B1

***** SEARCH POINT *** 100 *****

FXNS CALLS... 398 GRAD CALLS... 101 RHO = 0.228736D-04

F = -0.800000D 00 AUG. LAG. = -0.800000D 00 GMAG = 0.152370D-03

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
-0.167760D-04 0.286410D-03 0.137263D-03

DFLX = 0.727485D-08 DELG = 0.352495D-03 RMAG = 0.318046D-03

X VALUES, X(1), ..., X(N)
0.600000D 00 0.800000D 00 -0.274592D-03

EQUALITY CONSTRAINT STATUS :
I HI(I) EMULT(I)
1 0.420722D-08 0.249984D 00

INEQUALITY CONSTRAINT STATUS :
J GI(J) OMULT(J)
1 0.394416D-08 0.299926D 00

W1 = 0.819200D 04 W2 = 0.819200D 04

* méthode de recherche d'une direction de descente :
DFP-SS avec réinitialisation

* paramètre de pénalisation de départ : $C_0 = 0.25$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 10^5$

Nombre de cycles : 15

Résultats du problème B2

***** SEARCH POINT *** 100 *****

FNXS CALLS... 398 GRAD CALLS... 101 RHO = 0.228736D-04

F = -0.800000D 00 AUG. LAG. = -0.800000D 00 GMAG = 0.152370D-03

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
-0.167760D-04 0.286410D-03 0.137263D-03

DELX = 0.727485D-08 DELG = 0.352495D-03 RMAG = 0.318046D-03

X VALUES, X(1), ..., X(N)
0.600000D 00 0.800000D 00 -0.274592D-03

EQUALITY CONSTRAINT STATUS :
I HI(I) EMULT(I)
1 0.420722D-08 0.249984D 00

INEQUALITY CONSTRAINT STATUS :
J GI(J) CMULT(J)
1 0.394416D-08 0.299986D 00

W1 = 0.819200D 04 W2 = 0.819200D 04

* méthode de recherche d'une direction de descente :

DFP-SS sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 0.25$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : 10^5

Nombre de cycles : 15

Résultats du problème B3

SEARCH POINT *** 31 *****
FXNS CALLS... 89 GRAD CALLS... 32 RHO = 0.602321D-03
F = -0.800000D 00 AUG. LAG. = -0.800000D 00 GMAG = 0.870880D-06
SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
-0.395082D-06 0.241299D-05 0.294154D-10
DELX = 0.147275D-02 DELG = 0.245288D-05 RMAG = 0.244512D-05
X VALUES, X(1), ..., X(N)
0.600000D 00 0.800000D 00 -0.588131D-10
EQUALITY CONSTRAINT STATUS :
I HI(I) EMULT(I)
1 0.419998D-09 0.250000D 00
INEQUALITY CONSTRAINT STATUS :
J GI(J) CMULT(J)
1 0.447364D-09 0.300000D 00
W1 = 0.512000D 03 W2 = 0.512000D 03

- * méthode de recherche d'une direction de descente :
DFP avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 0.25$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{MAX} = 10^5$

Nombre de cycles : 11

Résultats du problème B4

***** SEARCH POINT *** 31 *****
FXNS CALLS... 29 GRAD CALLS... 32 RHO = 0.602321D-03
F = -0.800000D 00 AUG. LAG. = -0.800000D 00 GMAG = 0.870880D-06
SEARCH DIRECTION TO THIS POINT, P(1), ..., P(N) NG = 1
-0.395082D-06 0.241299D-05 0.294154D-10
DELX = 0.147275D-08 DELG = 0.245288D-05 RMAG = 0.244512D-05
X VALUES, X(1), ..., X(N)
0.670000D 00 0.800000D 00 -0.588131D-10
EQUALITY CONSTRAINT STATUS :
I HI(I) EMULT(I)
1 0.419998D-09 0.250000D 00
INEQUALITY CONSTRAINT STATUS :
J GI(J) OMULT(J)
1 0.447364D-09 0.300000D 00
W1 = 0.512000D 03 W2 = 0.512000D 03

- * méthode de recherche d'une direction de descente :
DFP sans réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 0.25$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{MAX} = 10^5$

Nombre de cycles : 11

Résultats du problème B5

***** SEARCH POINT *** 25 *****
FXNS CALLS... 65 GRAD CALLS... 26 RHO = 0.191085D-01
F = -0.800000D 00 AUG. LAG. = -0.800000D 00 GMAG = 0.958504D-06
SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 2
-7.174866D-04 0.216840D-05 0.101694D-10
DELX = 0.368800D-06 DELG = 0.215661D-04 RMAG = 0.193003D-04
X VALUES, X(1), ..., X(N)
0.670000D 00 0.800000D 00 -0.201109D-10
EQUALITY CONSTRAINT STATUS :
I HI(I) EMULT(I)
1 0.135513D-07 0.250000D 00
INEQUALITY CONSTRAINT STATUS :
J GI(J) GMULT(J)
1 -0.137809D-07 0.300000D 00
W1 = 0.320000D 02 W2 = 0.320000D 02

- * méthode de recherche d'une direction de descente :
DFP sans réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 0.5$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{MAX} = 10^5$

Nombre de cycles : 6

Résultats du problème B6

***** SEARCH POINT *** 35 *****

FXNS CALLS... 111 GRAD CALLS... 36 RHO = 0.111008D 01

F = -0.800000D 00 AUG. LAG. = -0.800000D 00 GMAG = 0.404682D-07

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 4
 -0.545078D-08 -0.160664D-08 0.202323D-06

DFLX = 0.224682D-06 DELG = 0.236434D-05 RMAG = 0.202402D-06

X VALUES, X(1), ..., X(N)
 0.670000D 00 0.800000D 00 -0.161350D-09

EQUALITY CONSTRAINT STATUS :

I	H(I)	EMULT(I)
1	0.760605D-10	0.250000D 00

INEQUALITY CONSTRAINT STATUS :

J	G(J)	CMULT(J)
1	-0.546436D-10	0.300000D 00

W1 = 0.256000D 03 W2 = 0.256000D 03

* méthode de recherche d'une direction de descente :

DFP sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 1$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $CMAX = 10^5$

Nombre de cycles : 8

Résultats du problème B7

***** SEARCH POINT *** 47 *****

FXNS CALLS... 157 GRAD CALLS... 48 RHO = 0.794349D-03

F = -0.800000D 00 AUG. LAG. = -0.800000D 00 GMAG = 0.536124D-07

SEARCH DIRECTION TO THIS POINT, R(1),...,R(N) NG = 2
-0.191951D-04 0.944126D-06 -0.254479D-09

DELX = 0.152660D-07 DELG = 0.193440D-04 RMAG = 0.192183D-04

X VALUES, X(1),...,X(N)
0.670000D 00 0.800000D 00 0.508736D-09

EQUALITY CONSTRAINT STATUS :
I HI(I) EMULT(I)
1 0.293492D-10 0.250000D 00

INEQUALITY CONSTRAINT STATUS :
J G1(J) CMULT(J)
1 -0.119651D-10 0.300000D 00

W1 = 0.102400D 04 W2 = 0.102400D 04

* méthode de recherche d'une direction de descente :

DFP sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 2$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 10^5$

Nombre de cycles : 9

Résultats du problème C1

***** SEARCH POINT *** 56 *****

FXNS CALLS... 189 GRAD CALLS... 57 RHO = 0.325052D-03

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.709674D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
-0.762472D-04 -0.759342D-04 -0.151734D-03

DELX = 0.004657D-07 DELG = 0.126033D-03 RMAG = 0.136018D-03

X VALUES, X(1), ..., X(N)
0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :
J GI(J) OMULT(J)
1 0.285077D-09 0.222222D 00

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K) VARIABLES	UPPER
	XMULTL(K)	G(K)-X(K) K X(K)-D(K)	MULTU(K)
	0.000000D 00	-0.133333D 01 1	
	0.000000D 00	-0.777778D 00 2	
	0.000000D 00	-0.444444D 00 3	

W1 = 0.000000D 00 W2 = 0.100000D 04

- * méthode de recherche d'une direction de descente :
DFP-SS avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 1$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{max} = 10^4$
- Nombre de cycles : 10

Résultats du problème C2

***** SEARCH POINT *** 56 *****

FXNS CALLS... 189 GRAD CALLS... 57 RHO = 0.325052D-03

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.709674D-06

SEARCH DIRECTION TO THIS POINT,R(1),...,R(N) NG = 1

-0.762472D-04 -0.759342D-04 -0.151734D-03

DELX = 0.004057D-07 DELG = 0.186033D-03 RMAG = 0.186018D-03

X VALUES,X(1),...,X(N)

0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :

J	G1(J)	MULT(J)
1	0.285077D-09	0.222222D 00

CONSTRAINED VARIABLE STATUS :

XMULTL(K)	LOWER	BOUNDED X(K) VARIABLES		UPPER	XMULTU(K)
		C(K)-X(K)	K		
0.000000D 00	-0.133333D 01	01	1		
0.000000D 00	-0.777778D 00	00	2		
0.000000D 00	-0.444444D 00	00	3		

W1 = 0.000000D 00 W2 = 0.100000D 04

* Méthode de recherche d'une direction de descente :

DFP-SS sans réinitialisation

* Paramètre de pénalisation de départ : $C_0 = 1$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : -10^4

Nombre de cycles : 7

Résultats du problème C3

***** SEARCH POINT *** 28 *****
FXNS CALLS... 88 GRAD CALLS... 29 RHO = 0.100000D-10
F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.598119D-07
SEARCH DIRECTION TO THIS POINT,R(1),...,R(N) NG = 1
-0.603921D-06 -0.538953D-06 -0.114257D-05
DELX = 0.140024D-16 DELG = 0.000000D 00 RMAG = 0.140024D-05
X VALUES,X(1),...,X(N)
0.133333D 01 0.777778D 00 0.444444D 00
INEQUALITY CONSTRAINT STATUS :
J G1(J) DMULT(J)
1 -0.434141D-08 0.222223D 00
CONSTRAINED VARIABLE STATUS :
LOWER BOUNDED X(K) VARIABLES UPPER
XMULT(K) C(K)-X(K) K X(K)-D(K) XMULT(K)
0.000000D 00 -0.133333D 01 1
0.000000D 00 -0.777778D 00 2
0.000000D 00 -0.444444D 00 3
W1 = 0.000000D 00 W2 = 0.128000D 03

* méthode de recherche d'une direction de descente :
DPP avec réinitialisation
* paramètre de pénalisation de départ : $C_0 = 1$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 7

Résultats du problème C4

```

***** SEARCH POINT *** 28 *****
FXNS CALLS... 38 GRAD CALLS... 29 RHO = 0.100000D-10
### F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.598119D-07
SEARCH DIRECTION TO THIS POINT,R(1),...,R(N) NG = 1
-0.605921D-06 -0.538953D-06 -0.114257D-05
DELX = 0.140024D-16 DELG = 0.000000D 00 RMAG = 0.140024D-05
X VALUES,X(1),...,X(N)
0.133333D 01 0.777778D 00 0.444444D 00
INEQUALITY CONSTRAINT STATUS :
J GI(J) OMULT(J)
1 -0.434141D-08 0.222223D 00
CONSTRAINED VARIABLE STATUS :
          LOWER      BOUNDED X(K) VARIABLES      UPPER
          XMULTL(K)   C(K)-X(K)      K      X(K)-D(K)      XMULTU(K)
0.000000D 00   -0.133333D 01      1
0.000000D 00   -0.777778D 00      2
0.000000D 00   -0.444444D 00      3
W1 = 0.000000D 00 W2 = 0.128000D 03
*****
    
```

* méthode de recherche d'une direction de descente : DFP sans réinitialisation
 * paramètre de pénalisation de départ : $C_0 = 1$
 facteur d'accroissement : $WF = 2$
 paramètre de pénalisation limite : $C_{MAX} = 10^4$
 Nombre de cycles : 7

***** A SMALLER VALUE OF THE AUGMENTED LAGRANGIAN COULD NOT BE
FOUND IN 4 CONSECUTIVE SEARCHES
YOU MAY WANT TO RERUN THE PROBLEM WITH A DIFFERENT STARTING POINT
AND/OR DIFFERENT SEARCH PARAMETERS AND/OR A DIFFERENT MODE OF SEARCH-
DIRECTION GENERATION

184

..... FINAL SEARCH POINT VALUES TO FOLLOW

Résultats du problème C5

***** SEARCH POINT *** 41 *****

FXNS CALLS... 100 GRAD CALLS... 41 RHO = 0.000000D 00

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.663608D-05

SEARCH DIRECTION TO THIS POINT, R(1),...,R(N) NG = 1
-0.277032E-05 -0.272429D-05 -0.537970D-05

DELX = 0.000000D 00 DELG = 0.000000D 00 RMAG = 0.663608D-05

X VALUES, X(1),...,X(N)
0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :
J G(J) OMULT(J)
1 0.175709D-08 0.222223D 00

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K) VARIABLES	UPPER
XMULT(K)	C(K)-X(K)	K	X(K)-D(K)
0.000000D 00	-0.133333D 01	1	
0.000000D 00	-0.777778D 00	2	
0.000000D 00	-0.444444D 00	3	

W1 = 0.000000D 00 W2 = 0.100000D 04

- * Méthode de recherche d'une direction de descente :
LFP avec réinitialisation
- * Paramètre de pénalisation de départ : $C_0 = 1$
facteur d'accroissement : $WF = 4$
paramètre de pénalisation limite : 10^4

Nombre de cycles : 11

Résultats du problème C6

***** SEARCH POINT *** 18 *****

FXNS CALLS... 52 GRAD CALLS... 19 RHO = 0.204816D-02

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.607445D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
 -0.620043D-05 -0.620064D-05 -0.124013D-04

DELX = 0.311081D-07 DELG = 0.150773D-04 RMAG = 0.151883D-04

X VALUES, X(1), ..., X(N)
 0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :
 J GI(J) OMULT(J)
 1 0.125198D-08 0.222222D 00

CONSTRAINED VARIABLE STATUS :

	LOWLR	BOUNDED X(K) VARIABLES	UPPLR
XMULTL(K)	C(K)-X(K)	K X(K)-D(K)	XMULTU(K)
0.000000D 00	-0.133333D 01	1	
0.000000D 00	-0.777778D 00	2	
0.000000D 00	-0.444444D 00	3	

W1 = 0.000000D 00 W2 = 0.160000D 03

* Méthode de recherche d'une direction de descente :
 DFP avec réinitialisation
 * paramètre de pénalisation de départ : $C_0 = 5$
 facteur d'accroissement : $WF = 2$
 paramètre de pénalisation limite : 10^4

Nombre de cycles : 5

A SMALLER VALUE OF THE AUGMENTED LAGRANGIAN COULD NOT BE
FOUND IN 4 CONSECUTIVE SEARCHES
YOU MAY WANT TO RERUN THE PROBLEM WITH A DIFFERENT STARTING POINT
AND/OR DIFFERENT SEARCH PARAMETERS AND/OR A DIFFERENT MODE OF SEARCH-
DIRECTION GENERATION

186

----- FINAL SEARCH POINT VALUES TO FOLLOW -----

Résultats du problème C7

***** SEARCH POINT *** 31 *****

FXNS CALLS... 170 GRAD CALLS... 28 RHO = 0.000000D 00

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.733951D 05

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
0.305133D 05 0.297074D 05 0.597767D 05

DELX = 0.000000D 00 DELS = 0.000000D 00 RMAG = 0.733951D 05

X VALUES, X(1), ..., X(N)
0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :
J GI(J) OMULT(J)
1 -0.177498D 08 0.222221D 00

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED	X(K) VARIABLES	UPPER	
XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)	XMULTU(K)	
0.000000D 00	-0.133333D 01	1			
0.000000D 00	-0.777778D 00	2			
0.000000D 00	-0.444444D 00	3			

W1 = 0.000000D 00 W2 = 0.100000D 04

* Méthode de recherche d'une direction de descente :

DFP avec réinitialisation

* paramètre de pénalisation de départ : $C_0 = 10$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 11

Résultats du problème C8

***** SEARCH POINT *** 85 *****

FXNS CALLS... 407 GRAD CALLS... 84 RHO = 0.250000D-11

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.690959D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
0.224081D-05 0.225915D-05 0.453747D-05

DELX = 0.138550D-16 DELG = 0.000000D 00 RMAG = 0.554199D-05

X VALUES, X(1), ..., X(N)
0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :
J G1(J) OMULT(J)
1 0.198059D-C8 0.222220D 00

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K) VARIABLES	UPPER	
XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)	XMULTU(K)
0.000000D 00	-0.133333D 01	1		
0.000000D 00	-0.777778D 00	2		
0.000000D 00	-0.444444D 00	3		

W1 = 0.000000D 00 W2 = 0.100000D 04

* méthode de recherche d'une direction de descente .

DPP avec réinitialisation

* paramètre de pénalisation de départ : $C_0 = 20$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 21

Résultats du problème 09

***** SEARCH POINT *** 80 *****

FXNS CALLS... 349 GRAD CALLS... 81 RHO = 0.250000D-11

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.326418D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
0.140846D-05 0.143759D-05 0.279428D-05

DELX = 0.860902E-17 DELG = 0.000000D 00 RMAG = 0.344361D-05

X VALUES, X(1), ..., X(N)
0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :
J GI(J) OMULT(J)
1 0.153820D-08 0.222221D 00

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K) VARIABLES	UPPER	
XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)	XMULTU(K)
0.000000D 00	-0.133333D 01	1		
0.000000D 00	-0.777778D 00	2		
0.000000D 00	-0.444444D 00	3		

W1 = 0.000000D 00 W2 = 0.100000D 04

- * méthode de recherche d'une direction de descente :
DFP avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 50$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 19

Résultats du problème C10

*****SEARCH POINT *** 61*****

FXNS CALLS... 337 GRAD CALLS... 61 RHO = 0.250000D-11

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.809717D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
0.107105D-05 0.107749D-05 0.220477D-05

DELX = 0.669384D-17 DELG = 0.000000D 00 RMAG = 0.267754D-05

X VALUES, X(1), ..., X(N)
0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :
J GI(J) OMJLT(J)
1 0.762723D-09 0.222221D 00

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED	X(K)	VARIABLES	UPPER	
	XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)		XMULTU(K)
	0.000000D 00	-0.133333D 01	1			
	0.000000D 00	-0.777778D 00	2			
	0.000000D 00	-0.444444D 00	3			

W1 = 0.000000D 00 W2 = 0.100000D 04

- * méthode de recherche d'une direction de descente :
DFP avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 100$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 19

Résultats du problème C11

***** SEARCH POINT *** 28 *****

FXNS CALLS... 84 GRAD CALLS... 29 RHO = 0.100000D-02

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.416877D-06

SEARCH DIRECTION TO THIS POINT, R(1),...,R(N) NG = 1
-0.515652D-06 -0.452720D-06 -0.972833D-06

DELX = 0.119049D-08 DELG = 0.362506D-06 RMAG = 0.119049D-05

X VALUES, X(1),...,X(N)
0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :
J GI(J) OMULT(J)
1 -0.101390D-07 0.222223D 00

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K)	VARIABLES	UPPER	
	XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)	XMULTU(K)
	0.000000D 00	-0.133333D 01	1		
	0.000000D 00	-0.777778D 00	2		
	0.000000D 00	-0.444444D 00	3		

W1 = 0.000000D 00 W2 = 0.500000D 02

- * méthode de recherche d'une direction de descente :
DFP avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 1$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : 50

Nombre de cycles : 7

CONVERGENCE CRITERION SATISFIED ##### 191
..... FINAL SEARCH POINT VALUES TO FOLLOW

Résultats du problème G12

***** SEARCH POINT *** 28 *****

FXNS CALLS... 88 GRAD CALLS... 29 RHO = 0.100000D-10

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.339186D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
-0.603921D-06 -0.538953D-06 -0.114257D-05

DELX = 0.140024D-16 DELG = 0.000000D 00 RMAG = 0.140024D-05

X VALUES, X(1), ..., X(N)
0.133333D 01 0.777778D 00 0.444444D 00

INEQUALITY CONSTRAINT STATUS :
J G1(J) OMULT(J)
1 -0.434141D-08 0.222223D 00

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K)	VARIABLES	UPPER	
	XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)	XMULTU(K)
	0.000000D 00	-0.133333D 01	1		
	0.000000D 00	-0.777778D 00	2		
	0.000000D 00	-0.444444D 00	3		

W1 = 0.000000D 00 W2 = 0.100000D 03

- * méthode de recherche d'une direction de descente :
DFP avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 1$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{MAX} = 100$

Nombre de cycles : 7

Résultats du problème C13

***** SEARCH POINT *** 16 *****

FXNS-CALLS... 43 GRAD-CALLS... 17 RHO = 0.204682D-02

F = 0.111111D 00 AUG. LAG. = 0.111111D 00 GMAG = 0.849286D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
-0.357964D-04 -0.357956D-04 -0.715898D-04

DELX = 0.179466D-06 DELG = 0.880592D-04 RMAG = 0.876802D-04

X VALUES, X(1), ..., X(N)
0.133333D 01 0.777778D 00 0.444445D 00

INEQUALITY CONSTRAINT STATUS :
J GI(J) OMULT(J)
1 0.182351D-08 0.222222D 00

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K)	VARIABLES	UPPER	
	XMULTL(K)	L(K)-X(K)	K	X(K)-D(K)	XMULTU(K)
	0.000000D 00	-0.133333D 01	1		
	0.000000D 00	-0.777778D 00	2		
	0.000000D 00	-0.444445D 00	3		

W1 = 0.000000D 00 W2 = 0.243000D 03

- * méthode de recherche d'une direction de descente :
DPP avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 1$
facteur d'accroissement : $WF = 3$
paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 5

Résultats du problème D1

***** SEARCH POINT *** 27 *****

FXNS CALLS... 90 GRAD CALLS... 28 RHO = 0.100000D-02

F = 0.266667D 01 AUG. LAG. = 0.266667D 01 GMAG = 0.335475D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
0.151613D-05 0.149377D-05

DELX = 0.212838D-08 DFLG = 0.213270D-05 RMAG = 0.212838D-05

X VALUES, X(1), ..., X(N)
0.100000D 01 -0.142560D-09

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K)	VARIABLES	UPPER	
	XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)	XMULTU(K)
	0.400000D 01	-0.297613D-09	1		
	0.100000D 01	0.142560D-09	2		

W1 = 0.000000D 00 W2 = 0.100000D 04

- * méthode de recherche d'une direction de descente :
DFP-SS avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 1$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 11

..... FINAL SEARCH POINT VALUES TO FOLLOW

Résultats du problème-D2

***** SEARCH POINT *** 27 *****

FXNS CALLS... 90 GRAD CALLS... 28 RHO = 0.100000D-02

F = 0.266667D 01 AUG. LAG. = 0.266667D 01 GMAG = 0.35475D-06

SEARCH DIRECTION TO THIS POINT,R(1),...,R(N) NG = 1
0.151613D-05 0.149377D-05

DELX = 0.212838D-08 DFLG = 0.213270D-05 RMAG = 0.212838D-05

X VALUES,X(1),...,X(N)
0.100000D 01 -0.142560D-09

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K) VARIABLES		UPPER	
XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)		XMULTU(K)
0.400000D 01	-0.297613D-09	1			
0.100000D 01	0.142560D-09	2			

W1 = 0.000000D 00 W2 = 0.100000D 04

* méthode de recherche d'une direction de descente :

DFP-SS sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 1$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 11

Résultats du problème D3

***** SEARCH POINT *** 25 *****

FNNS CALLS... 76 GRAD CALLS... 26 RHO = 0.1000000-02

F = 0.266667D 01 AUG. LAG. = 0.266667D 01 GMAG = 0.885756D-06

SEARCH DIRECTION TO THIS POINT,R(1),...,R(N) NG = 1
0.283206D-05 0.130750D-06

DELX = 0.283508D-08 DELG = 0.284639D-05 RMAG = 0.283508D-05

X VALUES,X(1),...,X(N)
0.1000000 01 0.132136D-11

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED	X(K)	VARIABLES	UPPER	
	XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)		XMULTU(K)
	0.4000000 01	-0.874427D-09	1			
	0.1000000 01	-0.132136D-11	2			

#1 = 0.000000 00 #2 = 0.100000 04

- * méthode de recherche d'une direction de descente :
DFP avec réinitialisation
- * paramètre de pénalisation de départ : $C_0 = 1$
facteur d'accroissement : $WF = 2$
paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 11

Résultats du problème D4

***** SEARCH POINT *** 25 *****

FXNS CALLS... 76 GRAD CALLS... 26 RHO = 0.100000D-02

F = 0.266667D 01 AUG. LAG. = 0.266667D 01 GMAG = 0.885756D-06

SEARCH DIRECTION TO THIS POINT, R(1),...,R(N) NG = 1
0.283206D-05 0.130750D-06

DELX = 0.283508D-08 DELG = 0.284639D-05 RMAG = 0.283508D-05

X VALUES, X(1),...,X(N)
0.100000D 01 0.132136D-11

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K)	VARIABLES	UPPER	
	XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)	XMULTU(K)
0.400000D 01		-0.874427D-09	1		
0.100000D 01		-0.132136D-11	2		

W1 = 0.000000D 00 W2 = 0.100000D 04

* méthode de recherche d'une direction de descente :

DFP sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 1$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 11

Résultats du problème D5

***** SEARCH POINT *** 20 *****

FXNS CALLS... 69 GRAD CALLS... 21 RHO = 0.100000D-02

F = 0.266667D 01 AUG. LAG. = 0.265667D 01 GMAG = 0.179714D-07

SEARCH DIRECTION TO THIS POINT,R(1),...,R(N) NG = 1
0.100225D-05 0.910375D-06

DELX = 0.135397D-08 DELG = 0.135694D-05 RMAG = 0.135397D-05

X VALUES,X(1),...,X(N)
0.100000D 01 0.123869D-11

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED	X(K)	VARIABLES	UPPER	
	XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)		XMULTU(K)
0.400000D 01	0.219376D-10		1			
0.100000D 01	-0.123869D-11		2			

W1 = 0.000000D 00 W2 = 0.100000D 04

* méthode de recherche d'une direction de descente :

DFP sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 1$

facteur d'accroissement : $WF = 4$

paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 8

Résultats du problème D6

***** SEARCH POINT *** 27 *****

FXNS CALLS... 115 GRAD CALLS... 27 RHO = 0.100000D-02

F = 0.266667D 01 AUG. LAG. = 0.266667D 01 GMAG = 0.931525D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
0.105937D-05 -0.277966D-07

DELX = 0.105973D-08 DELC = 0.106397D-05 RMAG = 0.105973D-05

X VALUES, X(1), ..., X(N)
0.100000D 01 -0.250979D-10

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED X(K)	VARIABLES	UPPER	
	XMULTI(K)	C(K)-X(K)	K	X(K)-D(K)	XMULTU(K)
	1.400000D 01	-0.956960D-09	1		
	0.100000D 01	0.250979D-10	2		

w1 = 0.000000D 00 w2 = 0.100000D 04

* méthode de recherche d'une direction de descente :

DFP sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 5$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 16

Résultats du problème D7

***** SEARCH POINT ** 21 *****

FXNS CALLS... 71 GRAD CALLS... 22 RHO = 0.100000D-02

F = 0.266667D 01 AUG. LAG. = 0.266667D 01 GMAG = 0.247418D-06

SEARCH DIRECTION TO THIS POINT, R(1), ..., R(N) NG = 1
0.100835D-05 -0.488010D-07

DFLX = 0.100953D-08 DELG = 0.779344D-06 RMAG = 0.100953D-05

X VALUES, X(1), ..., X(N)
0.100000D 01 0.226233D-10

CONSTRAINED VARIABLE STATUS :

	LOWER	BOUNDED	X(K) VARIABLES	UPPER	
XMULTL(K)	C(K)-X(K)	K	X(K)-D(K)	XMULTU(K)	
0.400000D 01	0.151756D-10	1			
0.100000D 01	-0.226233D-10	2			

W1 = 0.000000D 00 W2 = 0.100000D 04

* méthode de recherche d'une direction de descente :

DFP sans réinitialisation

* paramètre de pénalisation de départ : $C_0 = 3$

facteur d'accroissement : $WF = 2$

paramètre de pénalisation limite : $C_{MAX} = 10^4$

Nombre de cycles : 9

Dans les pages qui suivent , on trouvera un listing des différentes sous-routines décrites sommairement dans le chapitre I ainsi qu'une table alphabétique de toutes les variables .

TABLE DES VARIABLES

- AL : lagrangien augmenté
- ALF : minimum de l'approximation quadratique locale du
lagrangien augmenté
- C : borne inférieure de X
- CON1 : $DLX' \cdot DLG$
- CON2 : $DLG' \cdot R$
- CON3 : $(GAL - DLG)' \cdot DLX$
- CON4 : $(GAL - DLG)' \cdot R$
- CON5 : 0.5 CON2
- D : borne supérieure de X
- DELG : norme de DLG
- DELX : norme de DLX
- DIF : variable intermédiaire dans le calcul de ALF
- DLG : $\nabla L_k - \nabla L_{k-1}$
- DLX : $x_k - x_{k-1}$
- DS : valeur intermédiaire du pas dans la recherche uni-
directionnelle
- DT : idem
- D1,D2,D3 : idem
- D21 : $D2 - D1$
- D31 : $D3 - D1$

EMULT : vecteur des multiplicateurs de Lagrange pour
les contraintes d'égalité

EPD : 0.1 EPS1

EPS1 : utilisé dans le test de convergence ($GMAG < EPS1$)

EPS2 : valeur initiale de EPV

EPS3 : utilisé dans le test de convergence et dans le test
d'ajustement ($DELX < EPS3$)

EPT : 10. EPS2

EPV : d'abord égal à EPS2
égal à EPS1 lorsqu'après une phase d'ajustement on
a obtenu $GMAG < EPT$

ETERM : norme de HI

F : fonction objective

FS : valeur intermédiaire de la fonction objective dans
la recherche unidirectionnelle

FLAG : HI'. EMULT

GAL : gradient du lagrangien augmenté

GAMA : - CON3/CON4

GE : gradient des contraintes d'égalité

GEW1 : variable intermédiaire dans le calcul du gradient
du lagrangien augmenté

GEW2 : idem

GF : gradient de la fonction objective

GI : vecteur des contraintes d'inégalité

GIN : gradient des contraintes d'inégalité

GIS : vecteur intermédiaire des contraintes d'inégalité
dans la recherche unidirectionnelle

GLAG : variable intermédiaire dans le calcul de gradient
du lagrangien augmenté

GMAG : norme du gradient du lagrangien augmenté

GN : gradient du lagrangien augmenté évalué en un point
intermédiaire dans la recherche unidirectionnelle

- GO : idem
- GT : vecteur intermédiaire utilisé dans CONGR
- H : matrice approximant la matrice Hessienne du lagrangien augmenté soit par la méthode DFP , soit par DFP-SS
- HI : vecteur des valeurs des contraintes d'égalité
- HIS : vecteur intermédiaire des contraintes d'égalité dans la recherche unidirectionnelle
- I : indice compteur
- IBSD : compteur des mauvaises directions de recherche , initialisé à 0 et remis à 0 après chaque phase d'ajustement
- ICONV : égal à 0 au début du programme
1 quand les conditions de convergence sont satisfaites
- IC1 : compteur des étapes dans la recherche unidirectionnelle (on en fait au maximum 10)
- ID : utilisé pour la lecture et l'impression d'un commentaire
- IEMULT : nombre de multiplicateurs des contraintes d'égalité non initialisés à 0
- IFAIL : 0 au début et chaque fois qu'un a trouvé une plus petite valeur du lagrangien augmenté incrémenté de 1 chaque fois que la recherche unidirectionnelle n'a pas produit de plus petite valeur du lagrangien augmenté ; le programme stoppe lorsque $IFAIL \geq 4$
- IGE : nombre de composantes constantes non nulles du gradient des contraintes d'égalité
- IGF : idem pour le gradient de la fonction objective
- IGI : idem pour le gradient des contraintes d'inégalité

IMAX : nombre maximum de recherches unidirectionnelles
INDX : indice compteur
IOMULT : nombre de multiplicateurs des contraintes d'inégalité qui n'ont pas été initialisés à 0
IRESET : vaut 1 si on choisit de réinitialiser la direction de descente à l'opposé du gradient après chaque cycle à condition qu'au moins N recherches unidirectionnelles aient été faites depuis la dernière réinitialisation
vaut 0 sinon
ISS : vaut 0 pour la méthode DFP
vaut 1 pour la méthode DFP-SS
ISTOP : initialisé à 0 , le programme s'arrête lorsqu'il vaut 1
IT : nombre d'appels de la sous-routine SEARCH depuis le dernier appel de la sous-routine OUTPUT
ITOT : nombre total d'appels de la sous-routine SEARCH
ITRN : indice compteur
IUP : vaut 0 au début de la sous-routine SEARCH
vaut 1 lorsque dans la sous-routine VALUE la recherche a produit une plus petite valeur du lagrangien augmenté
IVAR : indice intermédiaire
IX : code employé pour déterminer si une variable est bornée ou non , et de quelle manière
IXMLTL : nombre de multiplicateurs des variables bornées inférieurement qui ne sont pas initialisés à 0
IXMLTL : idem pour les variables bornées supérieurement
I1 , I2 : indices compteurs
J,K : idem

KSRCH : nombre de recherches unidirectionnelles depuis
la dernière phase d'ajustement sauf si IBSD = 2 ;
dans ce cas KSRCH = N

L : indice compteur

N : nombre de composantes de X

NBV : nombre de variables bornées

NE : nombre de contraintes d'égalité

NEPI : nombre total de contraintes

NFE : compteur des appels de la sous-routine FXNS

NG : vaut 1 si la direction de recherche dans DFPRV est
l'opposé du gradient
sinon , NG est incrémenté de 1

NGE : nombre total d'appels de la sous-routine GRAD

NI : nombre de contraintes d'inégalité

NN : N.N

NPRINT : la sous-routine OUTPUT est appelée lorsque
IT = NPRINT

NSRCH : nombre maximum de recherches unidirectionnelles
à l'intérieur d'un cycle

NT : indice intermédiaire

NUP : s'il vaut 1 , on imprime des informations après
chaque ajustement

OMULT : vecteur des multiplicateurs des contraintes d'iné-
galité

OTERM : variable intermédiaire dans le calcul du lagran-
gien augmenté

R : direction de descente

RHO : valeur optimale du pas dans la direction de descente

RMAG : norme de R

SLOPE : R'.GAL

TEMP : variable intermédiaire utilisée dans le calcul du lagrangien augmenté et de son gradient

TFM : variable intermédiaire utilisée dans DFPRV

TP1 , TP2 : variables intermédiaires utilisées dans DMIN

VL : C - X

VU : X - D

WF : facteur d'accroissement du paramètre de pénalisation

W1 : paramètre de pénalisation pour les contraintes d'égalité

W1MAX : paramètre de pénalisation limite pour les contraintes d'égalité

W2 : paramètre de pénalisation pour les contraintes d'inégalité

W2MAX : paramètre de pénalisation limite pour les contraintes d'inégalité

W1D2 : $W1/2$

W2D2 : $1/(2 W2)$

X : la variable du problème

XMULTL : multiplicateur pour les variables bornées inférieurement

XMULTU : idem pour les variables bornées supérieurement

XN,XO : points intermédiaires dans la recherche unidirectionnelles

XT : idem

Y,YS,Y1,Y2,Y3 : valeurs intermédiaires du lagrangien augmenté dans la recherche unidirectionnelle

```

1      SUBROUTINE LPNLP(C,D,DLX,DLG,HI,HIS,CI,GIS,GAL,GE,GF,GIN,GO,H,
2      1 EMULT,OMULT,XMULTL,XMULTU,IX,R,X,XT)
3
4 C    COMMAND PROGRAM ; ORGANIZATION AND OPERATION OF ALGORITHM
5
6      IMPLICIT REAL*8(A-H,O-Z)
7      DIMENSION C(1),D(1),DLX(1),DLG(1),HI(1),HIS(1),GI(1),GIS(1)
8      DIMENSION GAL(1),GE(1),GF(1),GIN(1),GO(1),X(1),XT(1)
9      DIMENSION H(1),EMULT(1),OMULT(1),XMULTL(1),XMULTU(1),IX(1),R(1)
10     COMMON/C1/N,NE,NI,NBV,NEPI/C2/NFE,NGE/C3/W1,W2
11     COMMON/C4/W1MAX,W2MAX,WF/C6/DELX,DELG
12     COMMON/C9/KSRCH,IBSD/C20/NUP
13     COMMON/C10/EPS1,EPS2,EPS3/C11/ISTOP,ITOT
14     COMMON/C12/NSRCH,IT,NPRINT,IMAX
15     COMMON/C13/RMAG,RHO,GMAG,SLOPE/C21/F,AL
16     COMMON/C17/DS,FS,YS/C18/NG,IUP,IFAIL
17     COMMON/C25/ISS,IRESET,NN
18
19 C    INITIALIZATION
20
21     CALL INITL(C,D,HI,GI,GAL,GE,GF,GIN,GO,EMULT,OMULT,
22     1 XMULTL,XMULTU,IX,X)
23     ICONV=0
24     IFAIL=0
25     EPD=0.1*EPS1
26     EPT=10.*EPS2
27     EPV=EPS2
28     IF(NE.EQ.0) GOTO3
29     DO 2 I=1,NE
30 2     HIS(I)=HI(I)
31 3     IF(NI.EQ.0) GOTO 5
32     DO 4 I=1,NI
33 4     GIS(I)=GI(I)
34 5     FS=F
35
36 C    GENERATE SEARCH DIRECTION R ; CONDUCT UNIDIRECTIONAL SEARCH ;
37 C    AND UPDATE SEARCH-POINT STATUS
38
39 10     IF(ITOT.EQ.IMAX) GOTO 80
40     CALL DFPRV(GAL,H,DLX,DLG,N,R)
41 14     CALL SEARCH(C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,X,XT,
42     1 R)
43     IF(RHO.EQ.0.) GOTO 15
44     CALL GRAD(XT,GF,GE,GIN)
45     NGE=NGE+1
46     CALL GALAG(C,D,HI,GI,GAL,GE,GF,GIN,EMULT,OMULT,XMULTL,XMULTU,IX,
47     1 XT)
48 15     CALL DELTA(XT,X,GAL,GO,DLX,DLG,N)
49
50 C    INCREMENT COUNTERS ; TEST FOR OUTPUT ; AND TEST FOR UPDATE

```

```

51      ITOT=ITOT+1
52      KSRCH=KSRCH+1
53      IT=IT+1
54      IF(IT.EQ.NPRINT.OR.RHO.EQ.0.) GOTO 110
55  20   IF(IUP.EQ.0) GOTO 24
56      IFAIL=0
57      IF(DELX.GT.EPS3) GOTO 28
58      IF(GMAG.LT.EPV) GOTO 40
59      IF(DELG.GT.EPV) GOTO 28
60      IF(NG.EQ.1) GOTO 26
61      IF(DELG.LT.EPD.AND.KSRCH.EQ.1) GOTO 80
62      GOTO 26
63  24   IFAIL=IFAIL+1
64      IF(IFAIL.EQ.4) GOTO 80
65      IF(NG.EQ.1.OR.RHO.EQ.0.) GOTO 40
66  26   IBSD=IBSD+1
67  28   IF(KSRCH.GT.N) GOTO 30
68      IF(IBSD.EQ.2) KSRCH=N
69      GOTO 10
70  30   IF(KSRCH.EQ.NSRCH) GOTO 40
71      IF(GMAG-EPV) 40,40,35
72  35   IF(IBSD.LT.3) GOTO 10
73      RHO=0.
74
75
76  C   UPDATE MULTIPLIERS AND PENALTY WEIGHTS
77
78  40   IF(NEPI.EQ.0) GOTO 70
79      IF(NUP.EQ.0) PRINT 1000,ITOT
80      CALL UPDATE(C,D,HI,GI,EMULT,OMULT,XMULTL,XMULTU,IX,X)
81      CALL AULAG(C,D,HI,GI,EMULT,OMULT,XMULTL,XMULTU,IX,X,F,AL)
82      CALL GALAG(C,D,HI,GI,GAL,GE,GF,GIN,EMULT,OMULT,XMULTL,XMULTU,
83  1 IX,X)
84      GMAG=0.
85      DO 60 I=1,N
86      GO(I)=GAL(I)
87  60   GMAG=GMAG+GAL(I)*GAL(I)
88      GMAG=SQRT(GMAG)
89      IF(NUP.EQ.0) PRINT 1010,F,AL,GMAG
90  70   KSRCH=0
91      IBSD=0
92      IF(GMAG.LT.EPT) EPV=EPS1
93
94  C   TEST : CONVERGENCE CRITERION
95
96      IF(GMAG.GT.EPS1.OR.DELX.GT.EPS3) GOTO 10
97  75   ICONV=1
98
99  C   STOP : CONVERGENCE OR ITERATION LIMIT FINAL SEARCH POINT INFORMATION
100

```

```

101 80   ISTOP=1
102     PRINT 1050
103     IF(ICONV.EQ.1) GOTO 90
104     IF(IFAIL.EQ.4) GOTO 95
105     IF(ITOT.NE.IMAX) GOTO 85
106     PRINT 1020
107     GOTO 100
108
109 85   PRINT 1070
110     PRINT 1080
111 90   PRINT 1030
112     GOTO 100
113 95   PRINT 1060
114     PRINT 1080
115 100  PRINT 1040
116 110  CALL OUTPUT(C,D,HI,GI,EMULT,OMULT,XMULTL,XMULTU,IX,X,R)
117     IT=0
118     IF(ISTOP.EQ.0) GOTO 20
119     RETURN
120
121 C   ***** FORMATS *****
122
123 1000  FORMAT(///,20X,'#####',
124 1     1 '#####',/,20X,'MULTIPLIERS AND PENALTY ',
125 1     1 'WEIGHTS UPDATED AT SEARCH POINT *',16,'**')
126 1010  FORMAT(/,20X,'### F = ',D13.6,3X,'AUG. LAG. = ',D13.6,3X,
127 1     1 'GMAG = ',D13.6,/,20X,'#####',
128 1     1 '#####')
129 1020  FORMAT(20X,'##### MAXIMUM ITERATION LIMIT ',
130 1     1 '#####')
131 1030  FORMAT(20X,'##### CONVERGENCE ',
132 1     1 'CRITERION SATISFIED #####')
133 1040  FORMAT(20X,'..... FINAL SEARCH POINT VALUES TO ',
134 1     1 'FOLLOW .....')
135 1050  FORMAT('1')
136 1060  FORMAT(20X,'##### A SMALLER VALUE OF THE AUGMENTED ',
137 1     1 'LAGRANGIAN COULD NOT BE ',/,20X,'FOUND IN 4 CONSECUTIVE ',
138 1     1 'SEARCHES')
139 1070  FORMAT(20X,'LACK OF ACCURACY OR POOR CONVERGENCE RATE DETECTED:')
140 1080  FORMAT(20X,'YOU MAY WANT TO RERUN THE PROBLEM WITH A ',
141 1     1 'DIFFERENT STARTING POINT',/,20X,'AND/OR DIFFERENT SEARCH ',
142 1     1 'PARAMETERS AND/OR A DIFFERENT MODE OF SEARCH-',/,20X,
143 1     1 'DIRECTION',/)
144     END

```

```

1 SUBROUTINE INITL(C,D,HI,GI,GAL,GE,GF,GIN,GO,EMULT,OMULT,XMULTL,
2 1 XMULTU,IX,X)
3
4 C READS DATA AND INITIALIZES ALGORITHM ;
5 C OUTPUTS INITIAL CONDITIONS
6
7 IMPLICIT REAL*8(A-H,O-Z)
8 DIMENSION C(1),D(1),HI(1),GI(1),GAL(1),GE(1),GF(1),GIN(1),GO(1)
9 DIMENSION EMULT(1),OMULT(1),XMULTL(1),XMULTU(1),IX(1),X(1),ID(18)
10 COMMON/C1/N,NE,NI,NBV,NEPI/C2/NFE,NGE/C3/W1,W2
11 COMMON/C4/W1MAX,W2MAX,WF/C9/KSRCH,IBSD/C20/NUP
12 COMMON/C10/EPS1,EPS2,EPS3/C11/ISTOP,ITOT/C21/F,AL
13 COMMON/C12/NSRCH,IT,NPRINT,IMAX/C13/RMAG,RHO,GMAG,SLOPE
14 COMMON/C25/ISS,IRESET,NN
15
16 C INITIAL CONDITIONS
17
18 RHO=0.
19 GMAG=0.
20 IBSD=0
21 ISTOP=0
22 IT=0
23 ITOT=0
24 KSRCH=0
25 NFE=1
26 NGE=1
27 READ 1170,(ID(I),I=1,18)
28 READ 1000,N,NE,NI,NBV,IMAX,NPRINT,NUP,NSRCH,ISS,IRESET
29 NN=N*N
30 READ 1000,IGF,IGE,IGI,IEMULT,IOMULT,IXMLTL,IXMLTU
31 READ 1010,EPS1,EPS2,EPS3
32 READ 1010,(X(I),I=1,N)
33 NEPI=NE+NI+NBV
34 IF(NEPI.NE.0) READ 1010,W1,W2,W1MAX,W2MAX,WF
35 PRINT 1160
36 PRINT 1180,(ID(I),I=1,18)
37 PRINT 1020
38 PRINT 1030,N,NE,NI,NBV
39 PRINT 1300,IGF,IGE,IGI,IEMULT,IOMULT,IXMLTL,IXMLTU
40 PRINT 1040,EPS1,EPS2,EPS3
41 PRINT 1050,IMAX,NPRINT
42 IF(NUP.EQ.1) PRINT 1210
43 IF(NUP.EQ.0) PRINT 1220
44 PRINT 1060,NSRCH
45 PRINT 1310,ISS,IRESET
46 IF(NEPI.EQ.0) GOTO 5
47 PRINT 1070,W1,W2,WF,W1MAX,W2MAX
48
49 C COST FUNCTION INITIALIZATION
50

```

```

51 5 DO 10 K=1,N
52 10 GF(K)=0.
53 IF(IGF.EQ.0) GOTO 20
54 READ 1090,(K,GF(K),L=1,IGF)
55 CALL CONGR(GF,GO,IX,N,1)
56 20 CALL FXNS(X,F,HI,GI)
57
58 C EQUALITY CONSTRAINT INITIALIZATION
59
60 PRINT 1190
61 IF(NE.EQ.0) GOTO 60
62 XMULTU(K)=0.
63 DO 30 I=1,NE
64 30 EMULT(I)=0.
65 K=N*NE
66 DO 40 I=1,K
67 40 GE(I)=0.
68 IF(IEMULT.NE.0) READ 1090,(I,EMULT(I),L=1,IEMULT)
69 PRINT 1200
70 DO 50 I=1,NE
71 50 PRINT 1150,I,HI(I),EMULT(I)
72 IF(IGE.EQ.0) GOTO 70
73 READ 1090,(I,GE(I),L=1,IGE)
74 CALL CONGR(GE,GO,IX,N,NE)
75 GOTO 70
76 60 PRINT 1140
77
78 C INEQUALITY CONSTRAINT INITIALIZATION
79
80 70 PRINT 1230
81 IF(NI.EQ.0) GOTO 110
82 DO 80 J=1,NI
83 80 OMULT(J)=0.
84 K=N*NI
85 DO 90 J=1,K
86 90 GIN(J)=0.
87 IF(IOMULT.NE.0) READ 1090,(J,OMULT(J),L=1,IOMULT)
88 PRINT 1240
89 DO 100 J=1,NI
90 100 PRINT 1150,J,GI(J),OMULT(J)
91 IF(IGI.EQ.0) GOTO 120
92 READ 1090,(J,GIN(J),L=1,IGI)
93 CALL CONGR(GIN,GO,IX,N,NI)
94 GOTO 120
95 110 PRINT 1140
96
97 C BOUNDED VARIABLE INITIALIZATION
98
99 120 PRINT 1250
100 IF(NBV.EQ.0) GOTO 180

```

```

101 DO 130 K=1,N
102 IX(K)=0
103 C(K)=0.
104 D(K)=0.
105 130 XMULTL(K)=0.
106 READ 1100,(K,IX(K),C(K),D(K),L=1,NBV)
107 IF(IXMLTL.NE.0) READ 1090,(K,XMULTL(K),L=1,IXMLTL)
108 IF(IXMLTU.NE.0) READ 1090,(K,XMULTU(K),L=1,IXMLTU)
109 PRINT 1260
110 DO 170 K=1,N
111 IVAR=IX(K)+1
112 GOTO (170,140,150,160),IVAR
113 140 PRINT 1270,XMULTL(K),C(K),K
114 GOTO 170
115 150 PRINT 1280,K,D(K),XMULTU(K)
116 GOTO 170
117 160 PRINT 1290,XMULTL(K),C(K),K,D(K),XMULTU(K)
118 170 CONTINUE
119 GOTO 190
120 180 PRINT 1140
121
122 C FORMULATE AUGMENTED LAGRANGIAN AND GRADIENT
123
124 190 CALL GRAD(X,GF,GE,GIN)
125 CALL AULAG(C,D,HI,GI,EMULT,OMULT,XMULTL,XMULTU,IX,X,F,AL)
126 CALL GALAG(C,D,HI,GI,GAL,GE,GF,GIN,EMULT,OMULT,XMULTL,XMULTU,
127 1 IX,X)
128 DO 200 K=1,N
129 GO(K)=GAL(K)
130 200 GMAG=GMAG+GAL(K)*GAL(K)
131 GMAG=SQRT(GMAG)
132 PRINT 1110
133 PRINT 1130,(X(I),I=1,N)
134 PRINT 1120,F,AL,GMAG
135 PRINT 1160
136 RETURN
137
138 C ***** FORMATS *****
139
140 1000 FORMAT (10I5)
141 1010 FORMAT(8D10.5)
142 1020 FORMAT(/,20X,'##### INITIAL VALUES ',
143 1 '#####')
144 1030 FORMAT(/,20X,4X,'N= ',15,8X,'NE= ',15,8X,'NI= ',15,8X,'NBV= ',15)
145 1040 FORMAT(20X,'EPS1= ',D10.3,10X,'EPS2= ',D10.3,10X,'EPS3= ',D10.3)
146 1050 FORMAT(/,20X,'MAXIMUM ITERATION LIMIT...',15,2X,
147 1 ' ; OUTPUT PRINTED EVERY',14,' ITERATION(S)')
148 1060 FORMAT(/,20X,15,' IS THE MAXIMUM NUMBER OF UNIDIRECTIONAL ',
149 1 'SEARCHES DURING EACH',1,20X,'MINIMIZATION PHASE (MULTIPLIERS ',
150 1 'AND WEIGHTS ARE HELD CONSTANT WITHIN',1,20X,'EACH SUCH PHASE).')

```

```
151 1070 FORMAT(/,20X,'INITIAL PENALTY WEIGHTS ARE ',2(F7.2,2X),/,20X,
152 1 'RESPECTIVELY, AND ARE UPDATED BY A FACTOR OF ',F7.2,
153 1 'UNTIL THEY EXCEED',/,20X,'VALUES OF ',2(D10.3,2X)),
154 1090 FORMAT(15,D15.5,15,D15.5,15,D15.5,15,D15.5,15,D15.5)
155 1100 FORMAT(15,15,D15.5,D15.5,15,15,D15.5,D15.5)
156 1110 FORMAT(/,20X,'X VALUES,X(1),...,X(N)')
157 1120 FORMAT(/,20X,'### F = ',D13.6,3X,'AUG. LAG. = ',D13.6,3X,
158 1 'GMAG = ',D13.6,/,20X,'#####',
159 1 '#####',
160 1130 FORMAT (20X,D13.6,1X,D13.6,1X,D13.6,1X,D13.6,1X,D13.6)
161 1140 FORMAT(20X,5X,'UNCONSTRAINED')
162 1150 FORMAT(20X,15,2(3X,D13.6))
163 1160 FORMAT('1')
164 1170 FORMAT (18A4)
165 1180 FORMAT(20X,18A4)
166 1190 FORMAT(/,20X,'EQUALITY CONSTRAINT STATUS')
167 1200 FORMAT(20X,2X,'I',9X,'H1(I)',12X,'EMULT(I)')
168 1210 FORMAT(20X,'WEIGHT/MULTIPLIERS UPDATE INFORMATION SUPPRESSED')
169 1220 FORMAT(20X,'WEIGHT/MULTIPLIER UPDATE INFORMATION IS PRINTED')
170 1230 FORMAT(/,20X,'INEQUALITY CONSTRAINT STATUS. ')
171 1240 FORMAT(20X,2X,'J',9X,'GI(J)',12X,'OMULT(J)')
172 1250 FORMAT(/,20X,'CONSTRAINED VARIABLE STATUS : ')
173 1260 FORMAT(20X,4X,'XMULTL(K)',9X,'C(K)',9X,'K',10X,'D(K)',9X,
174 1 'XMULTU(K)')
175 1270 FORMAT(20X,D13.6,4X,D13.6,4X,I3)
176 1280 FORMAT(20X,34X,I3,4X,D13.6,4X,D13.6)
177 1290 FORMAT(20X,D13.6,4X,D13.6,4X,I3,4X,D13.6,4X,D13.6)
178 1300 FORMAT(/,20X,3X,'IGF',5X,'IGE',5X,'IGI',3X,'IEMULT',2X,'IOMULT',
179 1 3X,'IXMLTL',3X,'IXMLTU',/,20X,7(15,3X),/)
180 1310 FORMAT(/,20X,'ISS = ',I1,10X,'IRESET = ',I1)
181 END
```

```
1 SUBROUTINE CONGR(GT,GO,IX,N,NT)
2
3 C PRINTS CONSTANT GRADIENT COMPONENTS
4
5 IMPLICIT REAL*8(A-H,O-Z)
6 DIMENSION GT(1),GO(1),IX(1)
7
8 PRINT 1000
9 DO 20 I=1,NT
10 INDX=0
11 I1=(I-1)*N+1
12 I2=I*N
13 DO 10 K=I1,I2
14 IF(GT(K).EQ.0.)GOTO 10
15 INDX=INDX+1
16 IX(INDX)=K
17 GO(INDX)=GT(K)
18 10 CONTINUE
19 IF(INDX.EQ.0) GOTO 20
20 PRINT 1010,(IX(L),GO(L),L=1,INDX)
21 20 CONTINUE
22 RETURN
23
24 C ***** FORMATS *****
25 1000 FORMAT(/,20X,'CONSTANT GRADIENT COMPONENTS OF THE FUNCTION(S)')
26 1010 FORMAT(20X,I4,D14.6,I4,D14.6,I4,D14.6,I4,D14.6)
27 END
```

```
1 SUBROUTINE DFPRV(GAL,H,DLX,DLG,N,R)
2
3 C COMPUTES SEARCH DIRECTION R, VIA DFP METHOD WITH RESETS OF
4 C R = GAL BASED UPON UPDATE CRITERION, ETC.
5
6 IMPLICIT REAL*8(A-H,O-Z)
7 DIMENSION GAL(1),DLX(1),DLG(1),R(1),H(1)
8 COMMON /C9/KSRCH,IBSD/C13/RMAG,RHO,GMAG,SLOPE
9 COMMON /C18/NG,IUP,IFAIL/C25/ISS,IRESET,NN
10 NG = NG + 1
11 IF(RHO.EQ.0.) GOTO 10
12 IF(KSRCH.EQ.0 .AND. IFAIL.GT.0) GOTO 10
13 IF(IRESET.EQ.0) GOTO 40
14 IF(KSRCH.NE.0.OR.NG.LE.N) GOTO 40
15 10 NG = 1
16
17 C R = GAL
18
19 SLOPE = 0.
20 DO 20 I = 1,NN
21 20 H(I) = 0.
22 INDX = 0
23 DO 30 I = 1,N
24 H(INDX+I) = 1.
25 R(I) = -GAL(I)
26 SLOPE = SLOPE + R(I)*R(I)
27 30 INDX = INDX + N
28 RMAG = SQRT(SLOPE)
29 RETURN
30
31 C R VIA DFP OR DFPSS
32
33 40 CON1 = 0.
34 CON2 = 0.
35 CON3 = 0.
36 CON4 = 0.
37 INDX = 0
38 DO 50 I = 1,N
39 TFM = 0.
40 DO 45 J = 1,N
41 45 TFM = TFM + H(INDX + J)*DLG(J)
42 R(I) = TFM
43 CON1 = CON1 + DLX(I) * DLG(I)
44 CON2 = CON2 + DLG(I)*R(I)
45 IF(ISS.EQ.0) GOTO 50
46 TEMP = GAL(I) - DLG(I)
47 CON3 = CON3 + TEMP*DLX(I)
48 CON4 = CON4 + TEMP*R(I)
49 50 INDX = INDX + N
50 IF(CON1.EQ.(0.).OR.CON2.EQ.0.) GOTO 10
```

```
51      IF(ISS.EQ.0.OR.CON4.EQ.0.) GOTO 56
52      DO 55 I = 1,N
53 55     DLG(I) = R(I)/CON2 - DLX(I)/CON1
54      CON5 = .5*CON2
55      GAMA = - CON3/CON4
56 56     INDX = 0
57      DO 70 K = 1,N
58      ITRN = (K-1)*N
59      DO 60 J = K,N
60      L = INDX + J
61      IF(ISS.EQ.0.OR.CON4.EQ.0.) GOTO 57
62      H(L) = (H(L) - R(K)*R(J)/CON2 + CON5*DLG(K)*DLG(J))*GAMA
63      1 -DLX(K)*DLX(J)/CON1
64      GOTO 58
65 57     H(L) = H(L) - DLX(K)*DLX(J)/CON1 - R(K)*R(J)/CON2
66 58     H(ITRN+K) = H(L)
67 60     ITRN = ITRN + N
68 70     INDX = INDX + N
69     INDX = 0
70     DO 80 I = 1,N
71     TFM = 0.
72     DO 75 J = 1,N
73 75     TFM = TFM + H(INDX + J) * GAL(J)
74     R(I) = -TFM
75 80     INDX = INDX + N
76
77 C     TEST FOR A GOOD R ; IF R BAD,SET R = GAL
78     RMAG = 0.
79     SLOPE = 0.
80     DO 100 I = 1,N
81     RMAG = RMAG + R(I)*R(I)
82 100    SLOPE = SLOPE + R(I)*GAL(I)
83     RMAG = SQRT(RMAG)
84     IF(SLOPE.LT.0.) RETURN
85     GOTO 10
86     END
```

```

1      SUBROUTINE SEARCH(C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,
2      1 IX,X,XT,R)
3
4 C    CONDUCTS SEARCH ALONG R TO FIND STEPSIZE RHO THAT MINIMIZES THE AUGMENTED
5 C    LAGRANGIAN AL = F(X+RHO*R)
6
7      IMPLICIT REAL*8(A-H,O-Z)
8      DIMENSION C(1),D(1),HI(1),HIS(1),GI(1),GIS(1),EMULT(1),OMULT(1)
9      DIMENSION XMULTL(1),XMULTU(1),IX(1),XT(1),R(1),X(1)
10     COMMON/C1/N,NE,NI,NBV,NEPI/C2/NFE,NGE/C3/W1,W2
11     COMMON /C13/RMAG,RHO,GMAG,SLOPE/C18/NG,IUP,IFAIL
12     COMMON /C9/KSRCH,IBSD/C17/DS,FS,YS/C21/F,AL
13
14 C    SAVE STARTING POINT (Y1,D1) , SET INITIAL STEP D2, EVALUATE Y2
15
16     DS=0.
17     YS=AL
18     D1=0.
19     IC1=0
20     Y1=AL
21     IUP=0
22     D2=1.
23     IF(NG.EQ.1) D2=.1
24     IF(KSRCH.EQ.0.AND.IFAIL.GT.0) D2=0.05
25     IF(RMAG.LT.200.) GOTO 16
26     D2=10./RMAG
27     IF(D2.GT.(.001)) GOTO 16
28     D2=.001
29 10   CALL VALUE(C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,X,XT,
30     1 R,F,D2,Y2)
31
32 C    UNIDIRECTIONAL SEARCH
33
34     CALL DMIN(D1,D2,D3,Y1,Y2,Y3,SLOPE,1,D3)
35     IF(Y2-Y1) 20,200,50
36 20   IF(D3.LE.0.) GOTO 200
37     IF(D3.GE.(.9*D2).AND.D3.LE.(1.1*D2)) GOTO 410
38 25   IF(D3.LE.(5.*D2))GOTO 340
39     IF(D3.GT.(100.*D2)) D3=100.*D2
40     CALL VALUE (C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,X,XT,
41     1 R,F,D3,Y3)
42     IF(Y3-Y2) 30,200,200
43 30   D2=D3
44     Y2=Y3
45     GOTO 300
46 50   IF(D3.GT.(0.2*D2)) GOTO 100
47     IF(D3.LT.(0.01*D2)) D3=0.01*D2
48     CALL VALUE(C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,X,XT,R,
49     1 F,D3,Y3)
50     D2=D3

```

```
51      Y2=Y3
52      IF(Y2-Y1) 300,200,60
53 60    IF(IC1.GE.10) GOTO 410
54      IC1=IC1+1
55      CALL DMIN (D1,D2,D3,Y1,Y2,Y3,SLOPE,1,D3)
56      GOTO 50
57 100   CALL VALUE (C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,X,XT,
58 1      R,F,D3,Y3)
59      IF(Y3-Y1) 350,410,120
60 120   D2=0.2*D3
61      IC1=IC1+1
62      CALL VALUE (C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,X,XT,
63 1      R,F,D2,Y2)
64      IF(Y2-Y1) 240,410,150
65 150   IF(IC1.GT.10) GOTO 240
66 160   D3=D2
67      Y3=Y2
68      GOTO 120
69 200   D3=5.*D2
70      IC1=IC1+1
71      CALL VALUE (C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,X,XT,
72 1      R,F,D3,Y3)
73      IF(Y3-Y2) 210,210,220
74 210   IF(IC1.GT.10) GOTO 240
75      D1=D2
76      Y1=Y2
77      D2=D3
78      Y2=Y3
79      GOTO 200
80 220   IF(IUP) 410,410,240
81 240   CALL DMIN(D1,D2,D3,Y1,Y2,Y3,SLOPE,2,RHO)
82      GOTO 400
83 300   CALL DMIN (D1,D2,D3,Y1,Y2,Y3,SLOPE,1,D3)
84      IF(D3.GT.(0.9*D2).AND.D3.LT.(1.1*D2)) GOTO 410
85      IF(D3.LE.0.) GOTO 200
86      IF(IC1.GE.10) GOTO 410
87      IC1=IC1+1
88      GOTO 25
89 340   CALL VALUE(C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,X,XT,
90 1      R,F,D3,Y3)
91      IF(Y3-Y2) 350,350,240
92 350   CALL DMIN(D1,D2,D3,Y1,Y2,Y3,SLOPE,2,RHO)
93      IF(RHO.GT.(.9*D3).AND.RHO.LT.(1.1*D3)) GOTO 410
94
95 C    BEST STEPSIZE RHO FOUND FOR THIS SEARCH
96
97 400   CALL VALUE (C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,X,XT,
98 1      R,F,RHO,AL)
99 410   RHO=DS
100     DO 420 I=1,N
```

```
101 420 XT(I)=X(I)+RHO*R(I)
102      IF(NE.EQ.0) GOTO 440
103      DO 430 I=1,NE
104 430 HI(I)=HIS(I)
105 440 IF(NI.EQ.0) GOTO 460
106      DO 450 I=1,NI
107 450 GI(I)=GIS(I)
108 460 F=FS
109      AL=YS
110      RETURN
111      END
```

```
1 SUBROUTINE VALUE (C,D,HI,HIS,GI,GIS,EMULT,OMULT,XMULTL,XMULTU,IX,
2 1 X,XT,R,F,DT,Y)
3
4 C EVALUATES AUGMENTED LAGRANGIAN Y, AT STEPSIZE D, ALONG SEARCH DIRECTION R ;
5 C SAVES POINT IF BETTER
6
7 IMPLICIT REAL*8(A-H,O-Z)
8 DIMENSION C(1),D(1),HI(1),HIS(1),GI(1),GIS(1),EMULT(1),OMULT(1)
9 DIMENSION XMULTL(1),XMULTU(1),IX(1),X(1),XT(1),R(1)
10 COMMON /C1/ N,NE,NI,NBV,NEPI/C2/NFE,NGE/C3/W1,W2
11 COMMON /C17/DS,FS,YS/C18/NG,IUP,IFAIL
12
13 DO 5 I=1,N
14 5 XT(I)=X(I)+DT*R(I)
15 CALL FXNS(XT,F,HI,GI)
16 NFE=NFE+1
17 CALL AULAG(C,D,HI,GI,EMULT,OMULT,XMULTL,XMULTU,IX,XT,F,Y)
18 IF(Y-YS) 10,15,60
19 10 IUP=1
20 15 IF(NE.EQ.0) GOTO 30
21 DO 20 I=1,NE
22 20 HIS(I)=HI(I)
23 30 IF(NI.EQ.0) GOTO 50
24 DO 40 I=1,NI
25 40 GIS(I)=GI(I)
26 50 FS=F
27 DS=DT
28 YS=Y
29 60 RETURN
30 END
```

```
1      SUBROUTINE DMIN(D1,D2,D3,Y1,Y2,Y3,SLOPE,K,ALF)
2
3 C   ASSUMES QUADRATIC FORM  $AL=A*(D1-ALF)**2+B*(D1-ALF)+C$ 
4 C   MINIMUM  $ALF=D1-B/(2*A)$  FOUND BY QUADRATIC FIT OF DATA :
5 C   (POINT/SLOPE, POINT) OR (3 POINTS)
6
7      IMPLICIT REAL*8(A-H,O-Z)
8      GOTO (10,20),K
9
10 C   QUADRATIC FIT ; PSP
11
12 10   TP1=-SLOPE*D2
13      DIF=Y2-Y1+TP1
14      IF(DIF.EQ.0.) GOTO 30
15      ALF=(.5*TP1*D2)/DIF
16      RETURN
17
18 C   QUADRATIC FIT ; 3P
19
20 20   D21= D2-D1
21      D31=D3-D1
22      TP1=D31*(Y2-Y1)
23      TP2=D21*(Y3-Y1)
24      DIF=TP1-TP2
25      IF(DIF.EQ.0.) GOTO 30
26      ALF=D1-.5*(D21*TP2-D31*TP1)/DIF
27      RETURN
28
29 C   ZERO DIVISOR
30
31 30   ALF=25.*D2
32      RETURN
33      END
```

```
1 SUBROUTINE DELTA(XN,XO,GN,GO,DLX,DLG,N)
2
3 C COMPUTES: NORM(XNEW-XOLD),NORM(GNEW-GOLD),
4 C NORM(GNEW)
5
6 IMPLICIT REAL*8(A-H,O-Z)
7 DIMENSION XN(1),XO(1),GN(1),GO(1),DLX(1),DLG(1)
8 COMMON/C6/DELX,DELG/C13/RMAG,RHO,GMAG,SLOPE
9
10 DELG = 0.
11 GMAG = 0.
12 DO 10 I = 1,N
13 DLX(I) = XN(I) - XO(I)
14 DLG(I) = GN(I) - GO(I)
15 DELG = DELG + DLG(I)*DLG(I)
16 GMAG = GMAG + GN(I) * GN(I)
17 XO(I) = XN(I)
18 10 GO(I) = GN(I)
19 DELX = RHO*RMAG
20 DELG = SQRT(DELG)
21 GMAG = SQRT(GMAG)
22 RETURN
23 END
```

```
1      SUBROUTINE UPDATE(C,D,HI,GI,EMULT,OMULT,XMULTL,XMULTU,IX,X)
2
3  C    MULTIPLIERS AND PENALTY WEIGHTS UPDATED BY UPDATE RULES
4
5      IMPLICIT REAL*8(A-H,O-Z)
6      DIMENSION C(1),D(1),HI(1),GI(1),EMULT(1),OMULT(1)
7      DIMENSION XMULTL(1),XMULTU(1),IX(1),X(1)
8      COMMON/C1/N,NE,NI,NBV,NEPI/C3/W1,W2
9      COMMON/C4/W1MAX,W2MAX,WF/C2O/NUP
10
11 C    EQUALITY CONSTRAINT MULTIPLIERS
12
13      IF(NE.EQ.0) GOTO 20
14      DO 10 I=1,NE
15          TEMP=EMULT(I)+W1*HI(I)
16          EMULT(I)=EMULT(I)+W1*HI(I)
17 10    CONTINUE
18
19 C    INEQUALITY CONSTRAINT MULTIPLIERS
20
21 20    IF(NI.EQ.0) GOTO 50
22      DO 40 J=1,NI
23          TEMP=OMULT(J)+W2*GI(J)
24          OMULT(J)=DMAX1(0.0D1,TEMP)
25 40    CONTINUE
26
27 C    BOUNDED VARIABLES
28
29 50    IF(NBV.EQ.0) GOTO 200
30      DO 110 K=1,N
31          IVAR=IX(K)
32          IF(IVAR.EQ.0) GOTO 110
33          IF(IVAR.EQ.1.OR.IVAR.EQ.3) GOTO 70
34 60    IF(IVAR.EQ.2.OR.IVAR.EQ.3) GOTO 90
35      GOTO 110
36
37 C    LOWER BOUND MULTIPLIERS
38
39 70    TEMP=XMULTL(K)+W2*(C(K)-X(K))
40      XMULTL(K)=DMAX1(0.0D1,TEMP)
41      GOTO 60
42
43 C    UPPER BOUND MULTIPLIERS
44
45 90    TEMP=XMULTU(K)+W2*(X(K)-D(K))
46      XMULTU(K)=DMAX1(0.0D1,TEMP)
47 110   CONTINUE
48
49 C    PENALTY WEIGHT UPDATE
50
```

```

51 200 IF(NE.EQ.0) GOTO 210
52     IF(W1.LT.W1MAX) W1=W1*WF
53     IF(W1.GT.W1MAX) W1=W1MAX
54 210 IF((NI+NBV).EQ.0) GOTO 300
55     IF(W2.LT.W2MAX) W2=WF*W2
56     IF(W2.GT.W2MAX) W2=W2MAX
57
58 C   PRINT UPDATE STATUS IF NUP=0
59
60 300 IF(NUP.EQ.1) RETURN
61     IF(NE.EQ.0) GOTO 310
62     PRINT 1000
63     PRINT 1040,(HI(I),I=1,NE)
64     PRINT 1010
65     PRINT 1040,(EMULT(I),I=1,NE)
66 310 IF(NI.EQ.0) GOTO 320
67     PRINT 1020
68     PRINT 1040,(GI(J),J=1,NI)
69     PRINT 1030
70     PRINT 1040,(OMULT(J),J=1,NI)
71 320 IF(NBV.EQ.0) GOTO 370
72     PRINT 1050
73     DO 360 K=1,N
74     IVAR=IX(K)+1
75     GOTO (360,330,340,350),IVAR
76 330 VL=C(K)-X(K)
77     PRINT 1060,XMULTL(K),VL,K
78     GOTO 360
79 340 VL=C(K)-X(K)
80     PRINT 1070,X,VU,XMULTU(K)
81     GOTO 360
82 350 VL=C(K)-X(K)
83     VU=X(K)-D(K)
84     PRINT 1080,XMULTL(K),VL,K,VU,XMULTU(K)
85 360 CONTINUE
86 370 PRINT 1090,W1,W2
87     RETURN
88
89 C   ***** FORMATS *****
90
91 1000 FORMAT(/,20X,'EQUALITY CONSTRAINT VALUES, HI(1),...,HI(NE)')
92 1010 FORMAT(/,20X,'EQUALITY CONSTRAINT MULTIPLIERS, EMULT(1),..., ',
93     1 'EMULT(NE)')
94 1020 FORMAT(/,20X,'INEQUALITY CONSTRAINT VALUES, GI(1),...,GI(NE)')
95 1030 FORMAT(/,20X,'INEQUALITY CONSTRAINT MULTIPLIERS, OMULT(1),..., ',
96     1 'OMULT(NI)')
97 1040 FORMAT(20X,D13.6,1X,D13.6,1X,D13.6,1X,D13.6,1X,D13.6)
98 1050 FORMAT(/,20X,13X,'LOWER',7X,'BOUNDED X(K) VARIABLES',7X,'UPPER',
99     1 /,20X,4X,'XMULTL(K)',6X,'C(K)-X(K)',7X,'K',',X(K)-D(K)',7X,
100    1 'XMULTU(K)')

```

```
101 1060 FORMAT(20X,D13.6,4X,D13.6,4X,I3)
102 1070 FORMAT(20X,34X,I3,4X,D13.6,4X,D13.6)
103 1080 FORMAT(20X,D13.6,4X,D13.6,4X,I3,4X,D13.6,4X,D13.6)
104 1090 FORMAT(/,20X,'W1= ',D13.6,5X,'W2= ',D13.6)
105      END
```

```
1 SUBROUTINE AULAG(C,D,HI,GI,EMULT,OMULT,XMULTL,XMULTU,IX,X,F,AL)
2
3 C FORMULATES AUGMENTED LAGRANGIAN -AL
4
5 IMPLICIT REAL*8(A-H,O-Z)
6 DIMENSION HI(1),GI(1),EMULT(1),OMULT(1),X(1)
7 DIMENSION C(1),D(1),XMULTL(1),XMULTU(1),IX(1)
8 COMMON/C1/N,NE,NI,NBV,NEPI/C3/W1,W2
9
10 IF(NEPI.EQ.0)GOTO 200
11 FLAG=0.
12 ETERM=0.
13 OTERM=0.
14 W1D2=W1/2.
15 IF(W2.EQ.0.) GOTO 5
16 W2D2=1./(2.*W2)
17
18 C EQUALITY CONSTRAINT COMPONENTS
19
20 5 IF(NE.EQ.0) GOTO 20
21 DO 10 I=1,NE
22 TEMP=HI(I)
23 FLAG=FLAG+EMULT(I)*TEMP
24 10 ETERM=ETERM+TEMP*TEMP
25
26 C INEQUALITY CONSTRAINT COMPONENTS
27
28 20 IF(NI.EQ.0) GOTO 50
29 DO 40 J=1,NI
30 TEMP=OMULT(J)+W2*GI(J)
31 TEMP=D*MAX1(0.,TEMP)
32 40 OTERM=OTERM+TEMP*TEMP-OMULT(J)*OMULT(J)
33
34 C BOUNDED VARIABLES
35
36 50 IF(NBV.EQ.0) GOTO 120
37 DO 110 K=1,N
38 IVAR=IX(K)
39 IF(IVAR.EQ.0) GOTO 110
40 IF(IVAR.EQ.1.OR.IVAR.EQ.3) GOTO 70
41 60 IF(IVAR.EQ.2.OR.IVAR.EQ.3) GOTO 90
42 GOTO 110
43
44 C LOWER BOUND COMPONENTS
45
46 70 TEMP=XMULTL(K)+W2*(C(K)-X(K))
47 TEMP=D*MAX1(0.,TEMP)
48 OTERM=OTERM+TEMP*TEMP-XMULTL(K)*XMULTL(K)
49 GOTO 60
50
```

```
51 C UPPER BOUND COMPONENTS
52
53 90 TEMP=XMULTU(K)+W2*(X(K)-D(K))
54 TEMP=DMAX1(0.001,TEMP)
55 OTERM=OTERM+TEMP*TEMP-XMULTU(K)*XMULTU(K)
56 110 CONTINUE
57
58 C AL EQUALS THE SUM OF THE COMPONENTS
59
60 120 AL=F+FLAG+W1D2*ETERM+W2D2*OTERM
61 RETURN
62 200 AL=F
63 RETURN
64 END
```

```

1 SUBROUTINE GALG(C,D,HI,GI,GAL,GE,CF,GIN,EMULT,OMULT,XMULTL,
2 1 XMULTU,IX,X)
3
4 C FORMULATES THE GRADIENT OF THE AUGMENTED LAGRANGIAN - GAL
5
6 IMPLICIT REAL*8(A-H,O-Z)
7 DIMENSION C(1),D(1),HI(1),GI(1),GAL(1),GE(1),GF(1),GIN(1)
8 DIMENSION EMULT(1),OMULT(1),XMULTL(1),XMULTU(1),IX(1),X(1)
9
10 COMMON/C1/N,NE,NI,NBV,NEPI/C3/W1,W2
11
12 IF(NEPI.EQ.0) GOTO 200
13 DO 110 K=1,N
14 GLAG=0.
15 GEW1=0.
16 GIW2=0.
17
18 C EQUALITY CONSTRAINT COMPONENTS
19
20 IF(NE.EQ.0) GOTO 20
21 INDX=K
22 DO 10 I=1,NE
23 GLAG=GLAG+EMULT(I)*GE(INDX)
24 GEW1=GEW1+HI(I)*GE(INDX)
25 10 INDX=INDX+K
26
27 C INEQUALITY CONSTRAINT COMPONENTS
28
29 20 IF(NI.EQ.0) GOTO 50
30 INDX=K
31 DO 40 J=1,NI
32 TEMP=OMULT(J)+W2*GI(J)
33 IF(TEMP.LE.0.) GOTO 40
34 GIW2=GIW2+TEMP*GIN(INDX)
35 40 INDX=INDX+N
36
37 C ROUNDED VARIABLES
38
39 50 IF(NBV.EQ.0) GOTO 110
40 IVAR=IX(K)
41 IF(IVAR.EQ.0) GOTO 110
42 IF(IVAR.EQ.1.OR.IVAR.EQ.3) GOTO 70
43 60 IF(IVAR.EQ.2.OR.IVAR.EQ.3) GOTO 90
44 GOTO 110
45
46 C LOWER BOUND COMPONENTS
47
48 70 TEMP=XMULTL(K)+W2*(C(K)-X(K))
49 IF(TEMP.LE.0.) GOTO 60
50 GIW2=GIW2-TEMP

```

```
51      GOTO60
52
53 C    UPPER BOUND COMPONENTS
54
55 90    TEMP=XMULTU(K)+W2*(X(K)-D(K))
56      IF(TEMP.LE.0.) GOTO 110
57      GIW2=GIW2+TEMP
58
59 C    GAL EQUALS THE SUM OF THE COMPONENTS
60
61 110   GAL(K)=GF(K)+GLAG+W1*GEW1+GIW2
62      RETURN
63 200   DO 210 K=1,N
64 210   GAL(K)=GF(K)
65      RETURN
66      END
```

```
1      SUBROUTINE OUTPUT(C,D,HI,GI,EMULT,OMULT,XMULTL,XMULTU,IX,X,R)
2
3 C    OUTPUTS GENERAL AND TERMINAL SEARCH POINT INFORMATION
4
5      IMPLICIT REAL*8(A-H,O-Z)
6      DIMENSION C(1),D(1),HI(1),GI(1),EMULT(1),OMULT(1)
7      DIMENSION XMULTL(1),XMULTU(1),IX(1),X(1),R(1)
8      COMMON/C1/N,NE,NI,NBV,NEPI/C2/NFE,NGE/C3/W1,W2
9      COMMON/C6/DELX,DELG/C11/ISTOP,ITOT/C21/F,AL
10     COMMON/C13/RMAG,RHO,GMAG,SLOPE/C18/NG,IUP,IFAIL
11
12 C   GENERAL SEARCH POINT INFORMATION
13
14     PRINT 1000,ITOT
15     PRINT 1010,NFE,NGE,RHO
16     PRINT 1050,F,AL,GMAG
17     IF(ISTOP.EQ.0.AND.RHO.EQ.0.) GOTO 110
18     PRINT 1030,NG
19     PRINT 1040,(R(I),I=1,N)
20     PRINT 1020,DELX,DELG,RMAG
21     PRINT 1060
22     PRINT 1040,(X(I),I=1,N)
23     IF(ISTOP.EQ.0) RETURN
24
25 C   TERMINAL INFORMATION
26
27     IF(NEPI.EQ.0) GOTO 100
28     IF(NE.EQ.0) GOTO 20
29     PRINT 1190
30     PRINT 1200
31     DO 10 I=1,NE
32 10   PRINT 1150,I,HI(I),EMULT(I)
33 20   IF(NI.EQ.0) GOTO 40
34     PRINT 1230
35     PRINT 1240
36     DO 30 J=1,NI
37 30   PRINT 1150,J,GI(J),OMULT(J)
38 40   IF(NBV.EQ.0) GOTO 90
39     PRINT 1250
40     PRINT 1100
41     DO 80 K=1,N
42     IVAR=IX(K)+1
43     GOTO(80,50,60,70),IVAR
44 50   VL=C(K)-X(K)
45     PRINT 1090,XMULTL(K),VL,K
46     GOTO 80
47 60   VU=X(K)-D(K)
48     PRINT 1070,K,VU,XMULTU(K)
49     GOTO 80
50 70   VL=C(K)-X(K)
```

```

51      VU=X(K)-D(K)
52      PRINT 1080,XMULTL(K),VL,K,VU,XMULTU(K)
53 80    CONTINUE
54 90    PRINT 1130,W1,W2
55 100   PRINT 1110
56      PRINT 1120
57      RETURN
58 110   PRINT 1260
59      RETURN
60
61 C     ***** FORMATS *****
62
63 1000  FORMAT(///,20X,'*****SEARCH POINT ***',I6,
64      1 ' *****')
65 1010  FORMAT(/,20X,'FXNS CALLS... ',I6,5X,'GRAD CALLS... ',I6,5X,
66      1 'RHO = ',D13.6)
67 1020  FORMAT(/,20X,'DELX = ',D13.6,6X,'DELG = ',D13.6,6X,'RMAG = ',
68      1 D13.6)
69 1030  FORMAT(/,20X,'SEARCH DIRECTION TO THIS POINT,R(1),...,R(N)',10X,
70      1 'NG = ',I4)
71 1040  FORMAT(20X,D13.6,1X,D13.6,1X,D13.6,1X,D13.6,1X,D13.6)
72 1050  FORMAT(/,20X,'### F = ',D13.6,3X,'AUG. LAG. = ',D13.6,3X,
73      1 'GMAG = ',D13.6)
74 1060  FORMAT(/,20X,'X VALUES,X(1),...,X(N)')
75 1070  FORMAT(20X,34X,I3,4X,D13.6,4X,D13.6)
76 1080  FORMAT(20X,D13.6,4X,D13.6,4X,I3,4X,D13.6,4X,D13.6)
77 1090  FORMAT(20X,D13.6,4X,D13.6,4X,I3)
78 1100  FORMAT(/,20X,13X,'LOWER',7X,'BOUNDED X(K) VARIABLES',7X,'UPPER',
79      1 /,20X,4X,'XMULTL(K)',6X,'C(K)-X(K)',7X,'K',7X,'X(K)-D(K)',10X,
80      1 'XMULTU(K)')
81 1110  FORMAT(/,20X,'*****',
82      1 '*****')
83 1120  FORMAT('1')
84 1130  FORMAT(/,20X,'W1 = ',D13.6,5X,'W2 = ',D13.6)
85 1150  FORMAT(20X,I5,2(3X,D13.6))
86 1190  FORMAT(/,20X,'EQUALITY CONSTRAINT STATUS : ')
87 1200  FORMAT(20X,2X,'I',9X,'HI(I)',12X,'EMULT(I)')
88 1230  FORMAT(/,20X,'INEQUALITY CONSTRAINT STATUS : ')
89 1240  FORMAT(20X,2X,'J',9X,'GI(J)',10X,'OMULT(J)')
90 1250  FORMAT(/,20X,'CONSTRAINED VARIABLE STATUS : ')
91 1260  FORMAT(/,20X,'SEARCH FAILED TO FIND A SMALLER VALUE OF THE ',
92      1 'AUGMENTED LAGRANGIAN')
93      END

```

Enoncé du théorème des fonctions implicites

Soient Ω_n , (respectivement Ω_m) un ensemble ouvert de R^n (respectivement de R^m).

Considérons m fonctions réelles g_i ($1 \leq i \leq m$) définies et continues dans $\Omega_n \times \Omega_m$:

$$(x,y) \mapsto g_i(x,y) = g_i(x_1, \dots, x_n; y_1, \dots, y_m) \quad (1 \leq i \leq m)$$

De plus supposons que :

- 1) $g_i(x_{01}, \dots, x_{0n}; y_{01}, \dots, y_{0m}) = 0$ pour $i = 1, \dots, m$
- 2) pour $i=1, \dots, m$ et pour $k=1, \dots, m$, la dérivée partielle $D_{y_k} g_i(x,y)$ existe

et est une fonction continue de $(x,y) \in \Omega_n \times \Omega_m$

- 3) le déterminant suivant, appelé Jacobien est différent de 0 au (x_0, y_0)

$$J(x,y) = \begin{vmatrix} D_{y_1} g_1(x,y) & \dots & D_{y_m} g_1(x,y) \\ \vdots & & \vdots \\ D_{y_1} g_m(x,y) & \dots & D_{y_m} g_m(x,y) \end{vmatrix}$$

Dans ces conditions, il existe une boule ouverte $B_r(x_0) \subset \Omega_n$ et un système de m fonctions réelles f_i :

$x \mapsto f_i(x_1, \dots, x_n)$ définies et continues dans $B_r(x_0)$ t.q.

$$\begin{cases} f_i(x_{01}, x_{02}, \dots, x_{0n}) = y_{0i} \\ g_i(x_1, \dots, x_n; f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)) = 0 \end{cases} \quad (I)$$

pour tout $x \in B_r(x_0)$ et pour $i=1, \dots, m$

De plus : $x \mapsto (f_1(x), f_2(x), \dots, f_m(x))$ est la seule application de $B_r(x_0)$ dans une boule ouverte $B_s(y_0) \subset \Omega_n$ qui vérifie les équations (I) pour tout $x \in B_r(x_0)$.

Si en plus , les m fonctions g_i sont continuellement différentiables dans $\Omega_n \times \Omega_m$, il existe une boule $B_\varepsilon(x_0)$ dans laquelle les fonctions $f_i (1 \leq i \leq m)$ sont continuellement différentiables.

APPENDICE BExplications concernant l'utilisation des théorèmes d'analyse convexe dans la preuve de la différentiabilité continue de la fonctionnelle dualePréliminaires

Soit $f : \mathbb{R}^m \rightarrow \mathbb{R}$

1) Définissons la fermeture de f :

$$(cl\ g)(u) = \sup_{(y, \mu^*) \in D} (y'u - \mu^*)$$

où $D = \{(y, \mu^*) : y \in \mathbb{R}^m \quad y'u - \mu^* \leq g(u) \text{ pour tout } u \in \mathbb{R}^m\}$

2) Le domaine effectif de f ($= \text{dom } f$) est l'ensemble des points où $f(x)$ est finie

3) Une fonction essentiellement strictement convexe est une fonction strictement convexe sur tous les sous-ensembles convexes de $\{x \mid \partial f(x) \neq \emptyset\} = \text{dom } \partial f$ où $\partial f(x)$ est le sous-différentiel de f au point x . Une fonction strictement convexe est donc essentiellement strictement convexe.

4) Définissons la fonction conjuguée f^* de f :

$$f^*(y) = \sup_u (y'u - g(u))$$

5) Une fonction essentiellement régulière est une fonction convexe propre qui satisfait :

a) $C = \text{int}(\text{dom } f) \neq \emptyset$

b) f est différentiable sur C

c) $\lim_{i \rightarrow \infty} |\nabla f(x_i)| = +\infty$

pour $\{x_i\}$ une suite de points de C avec $x_i \rightarrow x$ un point de la frontière de C
 $i \rightarrow \infty$

Théorème 7.4 : la fermeture d'une fonction strictement convexe est strictement convexe sur son domaine effectif.

Théorème 26.3 : une fonction convexe propre fermée est essentiellement strictement convexe si et seulement si sa conjuguée est essentiellement régulière.

Corollaire 25.5.1 : Si une fonction convexe propre finie sur un ouvert convexe est différentiable alors elle est continuellement différentiable.

Démonstration

Considérons la fonction $g(u)$ qui prend la valeur :

$$g(u) = \begin{cases} p_c(u) & \text{sur } B(0, \rho) \\ + \infty & \text{ailleurs} \end{cases}$$

Cette fonction est strictement convexe.

Considérons maintenant sa fermeture : d'après le th.7.4. , celle-ci est également strictement convexe sur son domaine effectif (qui est la boule fermée $\bar{B}(0, \rho)$) donc d'après 3) la fermeture est essentiellement strictement convexe.

Considérons maintenant sa fonction conjuguée :

- celle-ci est partout finie (voir page 59)
- d'après le th.26.3 , elle est essentiellement régulière et donc possède les trois propriétés a), b) et c) et en particulier comme $C = \mathbb{R}^m$, elle est différentiable sur \mathbb{R}^m . Il reste à remarquer que le cor.25.5.1 assure qu'elle différentiable continuellement.

APPENDICE CLa méthode des multiplicateurs avec paramètre de pénalisation dépendant de x

Considérons un problème de minimisation avec contraintes de la forme :

$$\begin{aligned} \min p_0(x) \\ p_i(x) \leq 0 \quad i=1, \dots, m \end{aligned} \quad (37)$$

où $p_0, p_i : \mathbb{R}^n \rightarrow \mathbb{R}$

Supposons que \bar{x} est une solution optimale de ce problème. Etudions la méthode suivante :

PAS 1 : $y_k \in \mathbb{R}^m$ et $c_k^i : \mathbb{R}^n \rightarrow \mathbb{R}$ étant donnés ,

minimiser $P(x, y_k, \tilde{c}_k^i(x)) =$

$$p_0(x) + \sum_{i=1}^m \frac{1}{2\tilde{c}_k^i(x)} \left\{ \left[\max(0, y_k^i + \tilde{c}_k^i(x)p_i(x)) \right]^2 - (y_k^i)^2 \right\} \quad (38)$$

PAS 2 : définir $c_{k+1}^i : \mathbb{R}^n \rightarrow \mathbb{R}$ et ajuster y_k par l'itération suivante :

$$y_{k+1}^i = \max\{0, y_k^i + \tilde{c}_k^i(x_k)p_i(x_k)\} \quad (39), \text{ où } x_k \text{ min (38)}$$

Retourner au pas 1.

$\tilde{c}_k^i : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction à valeurs strictement positives pour $i=1, \dots, m$ et $k=1, \dots$
et

$$\tilde{c}_{k+1}^i(x) \geq \tilde{c}_k^i(x) \text{ pour tout } x$$

(par exemple $\tilde{c}_k^i(x) = c_k^i r_i(x)$ où $\{c_k^i\}$ $i=1, \dots, m$ est une suite croissante de scalaires strictement positifs et $r_i(x)$ une fonction à valeurs strict. positives. Pour cette méthode on a le résultat suivant :

Lemme 1

Supposons que le problème (37) vérifie les hypothèses (A') et (B) de la première partie.

Dans une boule $B(\bar{x}, \epsilon)$ on fait les hypothèses suivantes :

$\tilde{c}_k^i : R^n \rightarrow R$ $i=1, \dots, m$ $k=1, 2, \dots$ est une fonction strictement positive

$\tilde{c}_{k+1}^i(x) \geq \tilde{c}_k^i(x)$ pour tout $x \in B(\bar{x}, \epsilon)$

\tilde{c}_k^i est continuellement différentiable dans la boule $B(\bar{x}, \epsilon)$ il existe un scalaire c^* strictement positif tel que $\tilde{c}_k^i(x) \geq c^*$ pour tout $x \in B(\bar{x}, \epsilon)$

Si x_k est une solution optimale du problème (38) et que y_k est ajusté par l'itération (39), la suite de points $\{(x_k, y_k)\}$ converge asymptotiquement vers le vecteur optimal (\bar{x}, \bar{y}) du problème (37).

De plus l'itération d'ajustement (39) accélère la convergence et on a pas besoin de faire croître la valeur de la fonction $\tilde{c}_k^i(x)$ indéfiniment quand k

Démonstration

Supposons donc que x_k est une solution optimale du problème (38).

Dans ce cas posons :

$$\gamma_k = \left\| \frac{1}{2} \sum_{i=1}^m \left[\max \left(-\frac{y_k^i}{c_k^i(x_k)}, p_i(x_k) \right) \right]^2 \nabla \tilde{c}_k^i(x) \right\| \quad (40)$$

Puisque $p_i(\cdot)$ est une fonction continue et que $c_k^i(\cdot)$ $i=1, \dots, m$ est continuellement différentiable, si (x_k, y_k) est suffisamment proche de (\bar{x}, \bar{y}) alors

$$\max \left(-\frac{y_k^i}{c_k^i(x_k)}, p_i(x_k) \right) \text{ est petit et on peut sup-}$$

poser que $\{\gamma_k\}$ est une suite bornée.

Considérons le problème suivant :

$$\min_{\tilde{x}} \tilde{P}(x, y_k, \tilde{c}_k^i(x_k)) =$$

$$\min_x p_0(x) + \sum_{i=1}^m \frac{1}{2\tilde{c}_k^i(x_k)} \left([\max\{0, y_k^i + c_k^i(x_k)p_i(x)\}]^2 - y_k^i{}^2 \right) \quad (41)$$

$\tilde{c}_k^i(x_k)$ étant une constante suffisamment grande, d'après la proposition 2 de la première partie, il existe un point unique \tilde{x}_k qui minimise approximativement le problème (41) ci-dessus de la façon suivante :

$$\nabla \tilde{P}(\tilde{x}_k, y_k, \tilde{c}_k^i(x_k)) = -\frac{1}{2} \sum_{i=1}^m \left(\max \left(-\frac{y_k^i}{\tilde{c}_k^i(x_k)}, p_i(x_k) \right) \right)^2 \nabla c_k^i(x_k) \quad (42)$$

en fait $\tilde{x}_k = x_k$

Calculons en effet le gradient de la fonction objective du problème (41) au point x_k :

$$\begin{aligned} \nabla \tilde{P}(x_k, y_k, c_k^i(x_k)) &= \nabla p_0(x_k) + \\ &\sum_{i=1}^m \frac{2\tilde{c}_k^i(x_k)}{2\tilde{c}_k^i(x_k)} \max\{0, y_k^i + c_k^i(x_k)p_i(x_k)\} \nabla p_i(x_k) \end{aligned}$$

Calculons aussi le gradient de la fonction objective du problème (38) qui est nul au point x_k :

$$\begin{aligned} \nabla P(x_k, y_k, \tilde{c}_k^i(x_k)) &= \nabla p_0(x_k) + \sum_{i=1}^m \frac{\tilde{c}_k^i(x_k)}{\tilde{c}_k^i(x_k)} \max\{0, y_k^i + \\ &+ \tilde{c}_k^i(x_k) p_i(x_k)\} \nabla p_i(x_k) + \sum_{i=1}^m \frac{p_i(x_k)}{\tilde{c}_k^i(x_k)} \max\{0, y_k^i + \\ &+ \tilde{c}_k^i(x_k) p_i(x_k)\} \nabla \tilde{c}_k^i(x_k) + \sum_{i=1}^m \frac{1}{-2(\tilde{c}_k^i(x_k))^2} [\max\{0, y_k^i + \\ &+ \tilde{c}_k^i(x_k) p_i(x_k)\}]^2 \nabla \tilde{c}_k^i(x_k) = 0 \end{aligned}$$

Or $\nabla P(x_k, y_k, \tilde{c}_k^i(x_k))$ peut encore s'écrire, d'après (43) :

$$\begin{aligned} \nabla P(x_k, y_k, \tilde{c}_k^i(x_k)) &= \nabla \tilde{P}(x_k, y_k, \tilde{c}_k^i(x_k)) + \\ &\sum_{i=1}^m \frac{p_i(x_k)}{\tilde{c}_k^i(x_k)} \max\{0, y_k^i + \tilde{c}_k^i(x_k) p_i(x_k)\} \nabla \tilde{c}_k^i(x_k) + \\ &\sum_{i=1}^m \frac{1}{-2(\tilde{c}_k^i(x_k))^2} \left[\left[\max\{0, y_k^i + \tilde{c}_k^i(x_k) p_i(x_k)\} \right]^2 - (y_k^i)^2 \right] \nabla \tilde{c}_k^i(x_k) = 0 \end{aligned}$$

1) si $y_k^i + \tilde{c}_k^i(x_k) p_i(x_k) \leq 0$ alors $\max\{0, y_k^i + \tilde{c}_k^i(x_k) p_i(x_k)\} = 0$

$$\begin{aligned} \text{donc } \nabla P(x_k, y_k, \tilde{c}_k^i(x_k)) &= \nabla \tilde{P}(x_k, y_k, \tilde{c}_k^i(x_k)) + \\ &+ \frac{1}{2} \sum_{i=1}^m \left(\frac{y_k^i}{\tilde{c}_k^i(x_k)} \right)^2 \nabla \tilde{c}_k^i(x_k) \end{aligned}$$

2) si $y_k^i + \tilde{c}_k^i(x_k) p_i(x_k) \geq 0$ alors après calculs on trouve :

$$\begin{aligned} \nabla P(x_k, y_k, \tilde{c}_k^i(x_k)) &= \nabla \tilde{P}(x_k, y_k, \tilde{c}_k^i(x_k)) + \\ &+ \frac{1}{2} \sum_{i=1}^m (p_i(x_k))^2 \nabla \tilde{c}_k^i(x_k) \end{aligned}$$

On peut rassembler 1) et 2) en écrivant :

$$\begin{aligned} \nabla P(x_k, y_k, \tilde{c}_k^i(x_k)) &= \nabla \tilde{P}(x_k, y_k, \tilde{c}_k^i(x_k)) + \\ &+ \frac{1}{2} \sum_{i=1}^m \left(\max \left\{ -\frac{y_k^i}{\tilde{c}_k^i(x_k)}, p_i(x_k) \right\} \right)^2 \nabla \tilde{c}_k^i(x_k) \end{aligned}$$

$$\begin{aligned} \text{d'où } \nabla P(x_k, y_k, \tilde{c}_k^i(x_k)) &= -\frac{1}{2} \sum_{i=1}^m \left(\max \left\{ \frac{-y_k^i}{\tilde{c}_k^i(x_k)}, p_i(x_k) \right\} \right)^2 \nabla \tilde{c}_k^i(x_k) \\ &= \nabla P(\tilde{x}_k, y_k, \tilde{c}_k^i(x_k)) \end{aligned}$$

Comme \tilde{x}_k est une solution unique $\tilde{x}_k = x_k$

On peut donc conclure que chaque point x_k minimisant (38) minimise aussi (41) approximativement avec une approximation égale à :

$$-\frac{1}{2} \sum_{i=1}^m \left(\max \left\{ \frac{-y_k^i}{\tilde{c}_k^i(x_k)}, p_i(x_k) \right\} \right)^2 \nabla \tilde{c}_k^i(x_k)$$

La proposition 2 de la première partie assure également qu'il existe un scalaire $M > 0$ tel que

$$\|x_k - \bar{x}\| \leq M \left[\left(\frac{\|y_k - \bar{y}\|}{\tilde{c}_k(x_k)} \right)^2 + \left(\frac{(\gamma_k)^2}{\tilde{c}_k(x_k)} \right) \right]$$

$$\|\tilde{y}_k(x_k) - \bar{y}\| \leq M \left[\left(\frac{\|y_k - \bar{y}\|}{\tilde{c}_k(x_k)} \right)^2 + \left(\frac{(\gamma_k)^2}{\tilde{c}_k(x_k)} \right) \right]$$

où $\tilde{y}_k^i(x_k) = \max\{0, y_k^i + \tilde{c}_k^i(x_k) p_i(x_k)\}$ et $\tilde{c}_k(x_k) = \min_i \tilde{c}_k^i(x_k)$

Le terme $\frac{\gamma_k}{\tilde{c}_k(x_k)} \rightarrow 0$

et la suite $\{\tilde{y}_k(x_k)\}$ converge vers \bar{y}

et la suite $\{x_k\}$ vers \bar{x}

Remarquons que d'après (39) $y_{k+1}^i = \tilde{y}_k^i(x_k)$

et donc l'itération (39) accélère la convergence.

On peut donc conclure que la méthode des multiplicateurs avec paramètre de pénalisation dépendant de x possède les mêmes avantages que la méthode des multiplicateurs telle que nous l'avons étudiée dans la première partie.

Références

1. D.P.Bertsekas "On Penalty and Multiplier Methods for constrained Minimization" SiamJ. Control and Optimization Vol 14 , no 2, February 1976, 216-235
2. D.P.Bertsekas "Nondifferentiable Optimization via Approximation" Mathematical Programming Study 3 (1975) 1-25
3. W.C.Davidon "Variable Metric Method for Minimization" Atomic Energy Commission Research and Development Report , ANL-5990 , 1959
4. J.Dieudonné "Eléments d'Analyse" Gauthier-Villars Paris - 1972
5. R.Fletcher and M.J.D.Powell "A Rapidly Convergent Descent Method for Minimization" The Computer Journal Vol7 p 163-168 , July 1973
6. A.V.Fiacco et G.P.McCormick "Non linear Programming : Sequential Unconstrained Minimization Techniques" Wiley New-York , 1968
7. M.R.Hestenes "Multiplier and gradient Methods" J.Optimization Theory Appl.4,1969,303-320
8. F.A.Lootsma "Boundary Properties of Penalty Functions for Constrained Minimization" Thesis University of Eindhoven , The Netherlands, 1970
9. W.Murray "An Algorithm for Constrained Minimization" in R.Fletcher "Optimization" Academic Press , London and New-York 247-258
10. S.S.Oren and Luenberger , "Self-Scaling Variable Metric (SSVM) Algorithms, Part I : Criteria And Sufficient Conditions for Scaling a Class of Algorithms" Management Science, Vol.20 , 845-862 , Jan.1974
11. J.M.Ortega et W.C.Rheinboldt "Iterative solution of Nonlinear Equations in Several Variables" Academic Press 1970 - à partir de la page 281
12. D.A.Pierre and M.J.Lowe "Mathematical Programming Via Augmented Lagrangians : An Introduction with Computer Programs" Addison-Wesley Publishing Company , Massachusetts 1975

- 13.E.Polyak "The Convergence of the Penalty Function Method" *Ž.Vyčisl.Mat.i Mat.Fiz.*,11(1971)pp3-11
- 14.M.J.D.Powell "A Method for nonlinear Constraints in Minimization Problems" in *Optimization* , R.Fletcher ed.,Academic Press, New-York, 1969, 283-298
- 15.R.T.Rockafellar : "New Applications of duality in Convex Programming" Seventh International Symposium on Mathematical Programming, The Hague 1970, published in *Proceedings of the Fourth Conference on Probability*, Brasov, Romania, 1971
- 16.R.T.Rockafellar : "Convex Analysis" Princeton University Press 1970
-

Table des matières

Introduction	3
<u>Première partie : Méthode des multiplicateurs</u>	
<u>Introduction</u>	7
a. Description de la méthode et présentation de l'analyse	7
b. Les hypothèses pour lesquelles l'analyse est valable	12
c. L'algorithme de la méthode de pénalisation généralisée	14
d. Interprétation de l'algorithme	14
<u>Chapitre I : la convergence de l'algorithme</u>	
A. Proposition 1	17
B. Interprétation de la proposition 1	17
C. Démonstration	22
<u>Chapitre II : convergence de l'algorithme dans le cas d'une minimisation non exacte du lagrangien augmenté</u>	40
A. Proposition 2	41
B. Interprétation	42
C. Démonstration	43
<u>Chapitre III : la méthode des multiplicateurs dans un cadre de dualité globale</u>	
A. Solution du problème non linéaire avec perturbation des contraintes	53
B. La fonctionnelle duale et ses propriétés	56
C. Calcul de la fonctionnelle duale et de ses dérivées partielles	64
D. Interprétation de la méthode des multiplicateurs dans le cadre de la dualité	68
<u>Chapitre IV : Traitement des contraintes d'inégalité</u>	
A. Validité de l'analyse pour les problèmes avec contraintes d'inégalité	69
B. Calcul du lagrangien augmenté, de l'itération des multiplicateurs et des critères d'arrêt	73

Deuxième partie : Application à des problèmes non différentiables

<u>Introduction</u>	80
a. Philosophie de la méthode d'approximation par un problème différentiable	80
b. Une forme d'approximation donnée par la méthode des multiplicateurs	81
c. Mesure de l'approximation	84
d. Description de la procédure d'approximation	90
e. L'ajustement des paramètres	90
f. Présentation de l'analyse	93
<u>Chapitre I : La convergence de la méthode</u>	95
<u>Chapitre II : Convergence des multiplicateurs et conditions d'optimalité</u>	105
<u>Chapitre III : L'itération des multiplicateurs accélère la convergence</u>	112

Troisième partie : L'implémentation de l'algorithme de la méthode des multiplicateurs

Introduction	131
L'implémentation de l'algorithme	132
L'utilisation du programme	141
Résultats numériques	152
<hr/>	
Table des variables	201
Listings des sous-routines	207
Appendice A :	232
Appendice B :	234
Appendice C :	236
Références	241