

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE.
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

UNIVERSITE MOULOUD MAMMERY, TIZI-OUZOU
Faculté des Sciences
Département de Mathématiques



**Mémoire de fin d'études en vue de
l'obtention du diplôme de MASTER
en Recherche Opérationnelle**

Thème

**La nouvelle méthode d'échange pour l'optimisation semi-infinie
convexe.**

Présenté par

AMROUCHI Zahia et CHALAL Sarah

Encadré par

Dr OUANES Mohand

Devant le jury d'examen composé de :

TALEB Youcef	M.A.A	UMMTO	Président
OUANES Mohand	M.C.A	UMMTO	Rapporteur
KARA Fadila	M.A.A	UMMTO	Examinatrice
KASDI Kamel	M.A.A	UMMTO	Examineur

Soutenu le 24/ 06/ 2013

Remerciements

Au terme de ce modeste travail, qu'il nous soit permis d'exprimer notre profonde gratitude à notre promoteur M.C.A OUANES Mohand, pour l'intérêt qu'il a porté à notre travail, par ses précieux conseils et ses critiques constructives.

. Grâce à lui, on a pu réaliser ce travail. on lui en est très reconnaissantes.

Nous remercions vivement l'ensemble des membres du jury pour l'honneur qu'ils nous font en acceptant de juger ce travail.

Encore et avec une grande fierté et honneur que nous tenons à présenter nos remerciements à toute la famille administrative du département des mathématiques, à tous ces enseignants qui nous ont permis d'acquérir des connaissances.

Nos remerciements chaleureux s'adressent également aux camarades et amis Belkhir Rafik et Nora,... qui nous ont beaucoup aidés dans ce travail.

Nous n'oublierons pas le grand mérite de nos chers parents qui nous ont soutenus et apporté un appui moral.

Enfin, que tous ceux et celles que nous avons involontairement oubliés et qui ont participé de près ou de loin à la réalisation de ce modeste travail, trouvent ici l'expression de notre gratitude.

Z. AMROUCHI et S. CHALAL.

Table des matières

Introduction	3
1 Rappels sur la convexité	7
1.1 Éléments d'analyse convexe	7
1.1.1 Ensembles convexes	7
1.1.2 Fonctions convexes	9
1.1.3 Programmes convexes	13
2 Optimisation	15
2.1 Optimisation sans contraintes	15
2.1.1 Conditions d'optimalité	16
2.1.2 Conditions nécessaires d'optimalité locale	16
2.1.3 Conditions suffisantes d'optimalité locale	17
2.1.4 Conditions d'optimalité pour les problèmes quadratiques	19
2.1.5 Méthodes itératives	20
2.1.6 La recherche linéaire d'un pas	23
2.2 Optimisation sous contraintes	30
2.2.1 Conditions nécessaires d'optimalités	31
2.2.2 Conditions nécessaires Karush-Kuhn et Tucker	33
2.2.3 Interprétation géométrique des conditions de Karush-Kuhn et Tucker	35
2.2.4 Conditions suffisantes d'optimalité	36
2.2.5 Méthodes itératives	38
3 Optimisation semi-infinie	47
3.1 Formulation d'un problème semi-infini:	47
3.2 Exemple pratique	48

4 Méthodes de résolution d'un problème semi-infini convexe	51
4.1 Résolution d'un problème semi-infini convexe par la nouvelle méthode d'échange	51
4.1.1 Problème semi-infini discrétisé	51
4.1.2 Principe de la méthode	52
4.1.3 Test d'optimalité	53
4.1.4 Algorithme de résolution de (S.I.C) par la nouvelle méthode d'échange	54
4.1.5 Exemples numériques	55
4.1.6 programmes sur MATLAB	58
4.2 Résolution d'un programme semi-infini convexe par la méthode des coupes	61
4.2.1 La résolution des programmes $(\check{P}_c)^k$	62
4.2.2 Teste d'optimalité de la solution Y^k du programme (\check{P}_c) pour le programme (P_c)	63
4.2.3 Construction de la coupe	65
4.2.4 Les coupes ne coupent jamais le domaine D	65
4.2.5 La convergence	66
4.2.6 Exemple numérique	68
4.2.7 programmes sur MATLAB	71
conclusion	73
Bibliographie	74

Table des figures

1.1	<i>Interprétation géométrique d'un ensemble convexe</i>	8
1.2	<i>Illustration de la définition (enveloppe convexe)</i>	9
1.3	<i>Courbe représentative d'une fonction convexe</i>	10
1.4	<i>Courbe représentative d'une fonction concave</i>	11
2.1	<i>Schéma général d'une méthode de descente</i>	23
2.2	<i>Interprétation géométrique de la méthode de Newton</i>	26
2.3	<i>Interprétation géométrique des conditions de Karush-Kuhn et Tucker</i>	35
2.4	<i>Illustration de la notion de point-col</i>	37
3.1	<i>Les robots humanoïdes permettent de valider les méthodes développées</i>	48

Introduction

Dans la vie moderne ,le désir humain de perfection évolue pour atteindre ce qui est meilleur grace à l'expression qu'il trouve dans la théorie de l'optimisation.cette théorie comprend l'étude quantitive des optimums et les méthodes pour les déterminer.

L'optimisation est un outil important en sciences appliquées et pour l'analyse des systèmes physiques.Elle cherche à améliorer une performance en se rapprochant d'un point optimal une fois qu'on a bien identifier l'objectif qui peut être le profit,le temps,l'énergie potentielle ou n'importe quelle quantité ou combinaison de qualité représenté par une valeur algébrique.Notre but est de trouver les valeurs des variables qui optimisent l'objectif.

Le processus qui permet l'identification des objectifs des problèmes donnés est appelé modélisation.Dés que le modèle a été formulé,un algorithme d'optimisation peut être utilisé pour la résolution du problème.

Il fallait attendre le milieu du vingtième siècle,avec l'émergence des calculateurs et surtout la fin de la seconde guerre mondiale pour voir apparaître des avancées ont été spectaculaires en termes de techniques d'otimisation.Ces avancées ont été essentiellement obtenues en Grande Bretagne.En 1947,Dantzig proposa un algorithme pour résoudre des problèmes linéaires.En 1975,Bellman énonça le principe d'optimisation des problèmes de programmation dynamique.

Les technologies actuelles cherchent de plus en plus à traiter des systèmes complexes, constitués par un grand nombre de paramètres liés les uns aux autres par une structure bien déterminée.

L'évolution générale comprend aussi la recherche de performance évoluées (notions de productivité,de coût,de qualité des produits...) et des performance optimales (aller sur la lune en consommant le minimum de carburant,planifier une économie de manière optimale,...)

Il n'existe pas un algorithme d'optimisation universel. Il existe plutôt beaucoup d'algorithmes adaptés à des types particuliers de problèmes d'optimisation. Il est alors laissé à la responsabilité de l'utilisateur le choix d'un algorithme qui soit approprié pour son application. Ce choix est fondamental, il peut déterminer le succès ou l'échec dans la recherche de la solution optimale et peut aussi influencer beaucoup le temps de calcul nécessaire à l'estimation de la solution.

Les applications sont nombreuses et multiples. Nous pouvons citer quelques exemples :

- La conception et la fabrication, aéronautique, aérospatiale.
- Les stratégies militaires.
- Le transport d'énergie des réseaux électrique.
- Les protocoles de transport d'information des réseaux informatique.
- L'analyse statistique de données issues de modèles expérimentaux.
- **La planification de trajectoires d'un robot.**
- La gestion de production (planing, gestion, ordonnancement).
- La pollution (minimiser le taux de pollution sur une surface d'air ou d'eaux).

L'optimisation est un ensemble de méthodes qui permet d'obtenir le meilleur résultat. Optimiser revient donc à estimer des minima et des maxima d'une fonction ou d'un système de fonctions.

Il s'agit, d'une part, de bien identifier la formulation en modèles d'optimisation. Et d'autre part, de présenter les techniques de résolution de ces problèmes. Beaucoup de méthodes ne sont valables que pour certains types de problèmes. Ainsi, il est important de bien connaître les caractéristiques du problème posé, afin d'identifier la technique appropriée pour sa résolution. Les problèmes d'optimisation sont classés en fonction des caractéristiques mathématiques de la fonction objectif, des contraintes et des variables d'optimisation. Les plus importantes classes d'optimisation sont :

- problèmes monovariante (une seule variable)
- problèmes multivariante (plus d'une variable)
- problème continue (variable réelles)
- problèmes mixte (variables réelles et entières)
- problèmes combinatoires (variables entières avec permutation)
- problèmes linéaires (fonction objectif linéaire)
- problèmes quadratique
- problèmes non linéaires

- problèmes avec contraintes
- problèmes sans contraintes

Les problèmes qui ont une infinité de contraintes et un nombre fini de variables, présentant des structures particulières de linéarité, convexité sont appelés **problèmes semi-infinis** .

Notre travail s'inscrit dans cette problématique, il consiste à étudier deux méthodes de résolution des problèmes semi-infinis convexes qui possèdent certaines particularités de structures, soit pour la fonction objectif, soit pour les contraintes.

La forme générale d'un problème semi-infini (p) est :

$$(P) \Leftrightarrow \begin{cases} f(x) \rightarrow \min, \\ g(x,s) \leq 0, \quad \forall s \in S; \\ x \in \mathbb{R}^n, \quad s \in \mathbb{R}^p, |S| = +\infty. \end{cases}$$

Les différents résultats aux quels nous avons aboutis sont exposés dans les chapitres constituant ce mémoire qui se décompose comme suit:

Chapitre 1: Rappels sur la convexité (Elements d'analyses convexes)

Dans ce chapitre nous donnons certaines généralités (définitions, propositions,...) sur la **convexité** et sa relation avec l'optimisation

Chapitre 2: Optimisation

Ce chapitre se décompose en deux parties essentielles:

*1^{ere} partie: Nous abordons **les conditions d'optimalité** des points extrêmes (minimums locaux, maximums globaux) ainsi que la description des algorithmes pour résoudre le problème d'optimisation sans contraintes. Nous étudions d'abord les méthodes de descente, ensuite les méthodes du gradient conjugué.

*2^{eme} partie: cette partie comporte les conditions d'optimalité avec méthodes de résolution pour l'optimisation avec contraintes à savoir les méthodes de pénalités (intérieur et extérieur) et la méthode de Lagrangian augmenté.

Chapitre 3: Optimisation semi-infini convexe

Ce chapitre est consacré à l'optimisation semi-infini convexe. Nous présentons la forme

générale d'un problème semi-infini et la modélisation d'un problème pratique qui consiste à optimiser la durée du mouvement d'un pas d'un robot humanoïde.

Chapitre 4: Les méthodes de résolution d'un problème semi-infini convexe

Nous présentons deux méthodes de résolution illustrées par des exemples numériques programmés avec MATLAB en utilisant la fonction **fseminf** :

- * *la nouvelle méthode d'échange* qui consiste à résoudre dans chaque une de ces itérations un sous problème fini en utilisant le théorème de K.K.T

- * *La méthode des coupes* qui nous permettra de résoudre un programme mathématique donné (**P**) linéaire ou convexe dont le domaine des solutions réalisables est D sur un ensemble plus grand H (en ajoutant des contraintes linéaires au fur et à mesure) qui ne coupent jamais le domaine D jusqu'à l'obtention de la solution optimale de problème donné (**P**)

et nous terminons par une conclusion.

Chapitre 1

Rappels sur la convexité

1.1 Éléments d'analyse convexe

Pour l'étude des problèmes d'optimisations, il est nécessaire de rappeler les notions spécifiques dont l'étude est basée sur l'analyse convexe. En effet la notion de convexité joue un rôle très important dans les problèmes d'optimisation avec ou sans contraintes, pour la plupart des algorithmes que nous décrirons, la convergence vers un optimum global ne pourra être démontrée qu'avec des hypothèses de convexité.

1.1.1 Ensembles convexes

Définition 1.1.1 (3). Un ensemble $S \subset \mathbf{R}^n$ est dit convexe si et seulement si:

$$\left\{ \begin{array}{l} \forall x \in S \\ \forall y \in S \\ \forall \lambda \in [0,1] \implies \lambda x + (1 - \lambda)y \in S \end{array} \right.$$

Autrement dit, S est convexe si et seulement si, pour deux points quelconques x et y pris dans S , le segment $[x,y]$ est tout entier contenu dans S . Une interprétation **optique** consiste à dire que dans une pièce convexe, deux personnes peuvent toujours s'apercevoir (voir figure 1.1).

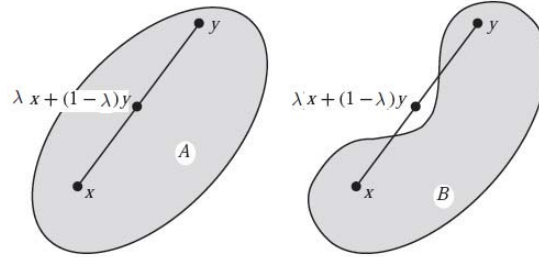


FIG. 1.1 – *Interprétation géométrique d'un ensemble convexe*

Définition 1.1.2. Combinaison convexe

Étant donné p points de \mathbf{R}^n (x^1, x^2, \dots, x^p) on dit que $x \in \mathbf{R}^n$ est combinaison convexe de ces points s'il existe des coefficients $\mu_1, \mu_2, \dots, \mu_p$ ($\mu_i \geq 0, \forall i = 1, 2, \dots, p$) tels que :

$$\sum_{i=1}^p \mu_i = 1 \quad \text{et} \quad X = \sum_{i=1}^p \mu_i x^i$$

On vérifie aisément qu'un ensemble $S \subset \mathbf{R}^n$ est convexe si et seulement si tout point, combinaison convexe de points de S , est dans S .

Preuve. $E = \{x \in \mathbf{R}^n / \sum_{i=1}^n \lambda_i = 1 \quad \text{et} \quad X = \sum_{i=1}^n \lambda_i x_i, \quad \lambda_i \geq 0\}$

soit $Y, Z \in E \quad \forall \alpha \in [0, 1]$.

on pose $Y = \sum_{i=1}^n \sigma_i x_i, \quad \sigma_i \geq 0, \quad \sum_{i=1}^n \sigma_i = 1$

et $Z = \sum_{i=1}^n \theta_i x_i, \quad \theta_i \geq 0, \quad \sum_{i=1}^n \theta_i = 1$

$\alpha Y + (1 - \alpha)Z = \alpha \sum_{i=1}^n \sigma_i x_i + (1 - \alpha) \sum_{i=1}^n \theta_i x_i$

$\sum_{i=1}^n (\alpha \sigma_i x_i + (1 - \alpha) \theta_i x_i)$

$\xi_i = \alpha \sigma_i + (1 - \alpha) \theta_i$

$\sum_{i=1}^n \xi_i = \sum_{i=1}^n (\alpha \sigma_i + (1 - \alpha) \theta_i)$

$\sum_{i=1}^n \xi_i = \alpha \sum_{i=1}^n \sigma_i + (1 - \alpha) \sum_{i=1}^n \theta_i = \alpha + (1 - \alpha) = 1$

$\Rightarrow \alpha Y + (1 - \alpha)Z \in E$

$\Rightarrow E$ est convexe.

Dans l'autre sens c'est évident.

Définition 1.1.3. Enveloppe convexe

Étant donné $S \subset \mathbf{R}^n$, on notera $\text{conv}(S)$ l'enveloppe convexe de S , c'est-à-dire l'ensemble des points de \mathbf{R}^n qui sont combinaison convexe de points de S .

D'une façon équivalente, S est convexe si et seulement si $S \equiv \text{conv}(S)$

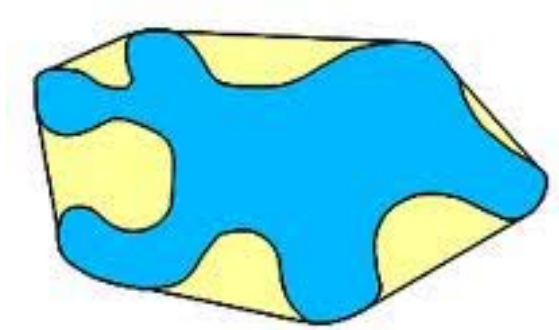


FIG. 1.2 – *Illustration de la définition (enveloppe convexe)*

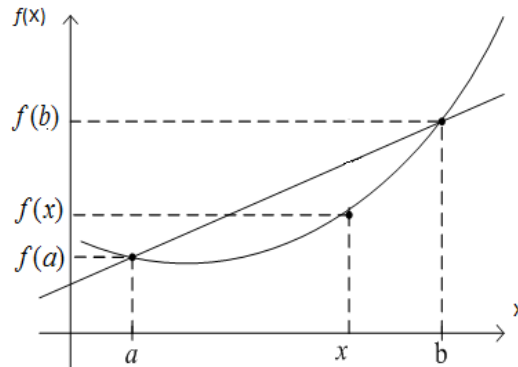
Proposition 1.1.1.

- L'intersection de sous-ensembles convexes est convexe.
- L'union de sous-ensembles convexes n'est pas convexe en général, mais l'union croissante de convexes (famille emboîtée) est convexe.

1.1.2 Fonctions convexes

Définition 1.1.4 (3). On dit qu'une fonction $f: \mathbf{R}^n \rightarrow \mathbf{R}$ définie sur S convexe, est convexe, si elle vérifie:

$$\{\forall x \in S, \forall y \in S, \forall \lambda \in [0,1], f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)\}$$

FIG. 1.3 – *Courbe représentative d'une fonction convexe*

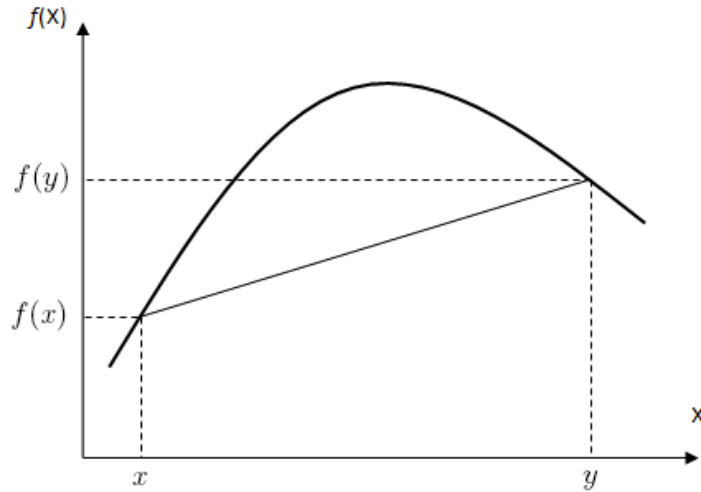
Le segment de droite reliant les points $(a, f(a))$ et $(b, f(b))$ se trouve totalement au-dessus du graphe de f . Le point $x = \lambda a + (1 - \lambda)b$ est situé quelque part entre a et b . Le point de coordonnées $(x, \lambda f(a) + (1 - \lambda)f(b))$ se trouve sur le segment de droite entre les points $(a, f(a))$ et $(b, f(b))$. Pour que la fonction soit convexe, il faut que ce point se trouve toujours (c'est-à-dire pour tout a, b et $0 < \lambda < 1$) au-dessus du graphe de la fonction.

Remarque 1.1.1. f est dite strictement convexe si l'ingalité stricte est toujours vérifiée pour $x \neq y$ et $\lambda \in [0, 1]$ $f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y)$

Définition 1.1.5 (3). "*Fonction concave*"

Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite concave si $(-f)$ est une fonction convexe, i.e. avec les mêmes notations, si

$$\forall x \in \mathbb{R}^n, \quad \forall y \in \mathbb{R}^n, \quad \forall \lambda \in [0, 1], \quad f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y) \quad (1.1)$$

FIG. 1.4 – *Courbe représentative d'une fonction concave*

Remarque 1.1.2. a) La convexité de la fonction est une caractéristique très importante en optimisation. En effet, lorsque la fonction n'est pas convexe, il est pratiquement impossible d'identifier un optimum global d'un problème d'optimisation par des méthodes classiques. Notons que l'importance de la convexité est liée aux problèmes de minimisation. Lorsque l'on étudie des problèmes de maximisation, on utilise la notion de concavité

b) Noter que convexité et concavité ne sont pas des propriétés complémentaires. Une fonction peut n'être ni convexe ni concave.

Définition 1.1.6. L'épigraphe

L'épigraphe d'une fonction f , noté $\text{épi}(f)$, est l'ensemble des vecteurs à $n+1$ composés (μ, x)

tels que $f(x) \leq \mu$: $\{(\mu, x) / f(x) \leq \mu, x \in \mathbb{R}^n, \mu \in \mathbb{R}\} \subset \mathbb{R}^{n+1}$.

Proposition 1.1.2. Critère de l'épigraphe

f est convexe si et seulement si son épigraphe est convexe, où

$$\text{épi}(f) = \{(x, r) \in A \times \mathbb{R}, f(x) \leq r\}$$

Preuve.

f convexe \Leftrightarrow son épigraphe est convexe

1. *f convexe \Rightarrow son épigraphe est convexe*

Soient $(x_1, r_1), (x_2, r_2) \in \text{épi}(f)$ et $\lambda \in [0, 1]$

On a $\lambda(x_1, r_1) + (1 - \lambda)(x_2, r_2) = (\lambda x_1 + (1 - \lambda)x_2, \lambda r_1 + (1 - \lambda)r_2)$

$$\begin{aligned} f(\lambda x_1 + (1 - \lambda)x_2) &\leq \lambda f(x_1) + (1 - \lambda)f(x_2) \quad \text{car } f \text{ est convexe} \\ &\leq \lambda r_1 + (1 - \lambda)r_2 \quad \text{car } (x_1, r_1), (x_2, r_2) \in \text{épi}(f) \end{aligned}$$

D'où $\lambda(x_1, r_1) + (1 - \lambda)(x_2, r_2) \in \text{épi}(f) \Rightarrow \text{épi}(f) \text{ est convexe} \quad (1)$

2. *épigraphe de f est convexe* \Rightarrow ? *f convexe*

Soient $(x_1, r_1), (x_2, r_2) \in \text{épi}(f)$ et $\lambda \in [0, 1]$

On a $\lambda(x_1, r_1) + (1 - \lambda)(x_2, r_2) = (\lambda x_1 + (1 - \lambda)x_2, \lambda r_1 + (1 - \lambda)r_2)$

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda r_1 + (1 - \lambda)r_2$$

On pose $r_1 = f(x_1)$ et $r_2 = f(x_2)$

D'où $f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \Rightarrow f \text{ est convexe} \quad (2)$

De (1) et (2) on a $f \text{ convexe} \Leftrightarrow \text{épigraphe convexe}$

Théorème 1.1.1.

Une combinaison linéaire à coefficients positifs de fonctions convexes est une fonction convexe.

Théorème 1.1.2.

Si f est continûment différentiable, les conditions (1) et (2) ci-dessous sont équivalentes; Si f est deux fois continûment différentiable, les conditions (1), (2) et (3) ci-dessous sont équivalentes :

1. f est convexe;
2. $\forall x \in \mathbb{R}^n, \forall x_0 \in \mathbb{R}^n, f(x) \geq f(x_0) + \nabla f^T(x_0) \cdot (x - x_0)$; c-à-d la courbe C de f est au dessus de ses tangentes.
3. $\forall x$, le hessien $\nabla^2 f(x)$ est une matrice semi-définie positive, c-à-d

$$\forall y, \quad y^T \cdot \nabla^2 f(x) \cdot y \geq 0. \quad (1.2)$$

corollaire 1: Une fonction quadratique dont le hessien est semi-défini positif est une fonction convexe.

1.1.3 Programmes convexes

Un programme mathématique dans \mathbf{R}^n est un problème d'optimisation sous contraintes de la forme:

$$(p) \left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{sous les contraintes :} \\ g_i(x) \leq 0 \quad \quad \quad i=1, \dots, m \\ x \in S \subset \mathbf{R}^n \end{array} \right.$$

Définition 1.1.7. On dit qu'un problème de programmation mathématique est convexe s'il consiste à minimiser une fonction convexe (resp: maximiser une fonction concave) sur un domaine convexe fermé. Ainsi, le problème (P) est un problème convexe si:

$$\left\{ \begin{array}{l} f(x) \text{ est convexe} \\ \text{les fonctions } g_i (i = 1, \dots, m) \text{ sont convexes} \\ S \subset \mathbf{R}^n \text{ est convexe fermé} \end{array} \right.$$

Théorème 1.1.3. Pour un programme convexe, tout optimum local est un optimum global.

Preuve. On peut toujours considérer un programme convexe de la forme

$$\left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{sous la contrainte :} \\ x \in S \end{array} \right.$$

Où f est une fonction convexe, et S un ensemble convexe.

Soit x^0 un optimum local. Pour montrer que x^0 est un minimum globale, considérons:

$y \in S$ ($y \neq x^0$) quelconque et montrons que $f(x^0) \leq f(y)$.

Raisonnons par l'absurde, en supposant que $f(x^0) > f(y)$.

En vertu de la convexité de f , ceci implique que:

$$\forall \lambda \in]0, 1[f(x^0 + \lambda(y - x^0)) \leq (1 - \lambda)f(x^0) + \lambda f(y) \leq f(x^0).$$

D'où il résulte une contradiction avec le fait que x^0 est un optimum local.

Donc, pour tout $y \in S$, on a bien $f(x^0) \leq f(y)$, ce que prouve que x^0 est un min global.

Chapitre 2

Optimisation

La programmation mathématique porte sur deux types d'optimisation :

- *l'optimisation sous contraintes* qui consiste à chercher $x^* \in S$ qui réalise le minimum (ou maximum) d'une fonction $f: \mathbf{R}^n \rightarrow \mathbf{R}$, liée à un ensemble d'équations ou d'inéquations linéaires ou non linéaires appelées contraintes
- *l'optimisation sans contraintes* qui tient à déterminer $x^* \in S$ qui sera le minimum (ou maximum) d'une fonction $f: \mathbf{R}^n \rightarrow \mathbf{R}$

2.1 Optimisation sans contraintes

Le problème à étudier ici est celui de la recherche du minimum (ou du maximum) d'une fonction réelle f de n variables réelles x_1, x_2, \dots, x_n . Usuellement f est appelée fonction objectif (ou critère, ou encore coût).

Soit $f: \mathbf{R}^n \rightarrow \mathbf{R}$ qui à tout $x \in \mathbf{R}^n, x = (x_1, x_2, \dots, x_n)^T$, associe la valeur réelle:

$$f(x) = f(x_1, x_2, \dots, x_n)$$

On cherche à résoudre:

$$\begin{cases} \text{Min} f(x) \\ x \in \mathbf{R}^n \end{cases}$$

Il s'agit donc de déterminer un point x^* de \mathbf{R}^n tel que :

$$\forall x \in \mathbf{R}^n : f(x^*) \leq f(x) \tag{2.1}$$

on parle d'un minimum global de f sur \mathbf{R}^n .

Une condition suffisante pour l'existence de $x^* \in \mathbf{R}^n$ vérifiant (2.1) est que f soit continue (ou, plus généralement semi-continue inférieurement) et possède la propriété de croissance à l'infinie, c'est-à-dire $f \rightarrow +\infty$ lorsque $\|x\| \rightarrow +\infty$

Le minimum global x^* est unique lorsque l'inégalité stricte: $f(x^*) < f(x)$ est vérifiée

$\forall x^* \in \mathbf{R}^n$, $x \neq x^*$ mais l'unicité n'est pas toujours garantie

Mais pour beaucoup de problèmes d'optimisation sans contraintes, les principales méthodes de résolution connues ne permettent pas la détermination d'un minimum global, il faut alors se contenter d'un minimum local, c'est -à-dire des points qui vérifient (2.1) seulement dans un voisinage de x^* . Nous allons voir maintenant comment de tels points peuvent être caractérisés.

2.1.1 Conditions d'optimalité

Avant de développer des algorithmes permettant de déterminer des solutions d'un problème d'optimisation, il faut être capable de décider si un point donné est optimal ou non. Ces conditions d'optimalité ont trois rôles essentiels dans le développement des algorithmes

1. elles procurent une analyse théorique du problème,
2. elles inspirent directement des idées pour les algorithmes,
3. elles permettent de déterminer un critère d'arrêt pour les algorithmes itératifs.

2.1.2 Conditions nécessaires d'optimalité locale

supposons que f est continue et a des dérivées partielles premières $\partial f / \partial x_i$ et secondes $\partial^2 f / \partial x_i \partial x_j$ continues pour tout $x \in \mathbf{R}^n$.

Théorème 2.1.1. [2] "Conditions nécessaires d'optimalité locale"

Une condition nécessaire pour que x^* soit un minimum local de f est :

- a) $\nabla f(x^*) = 0$
- b) si f est deux fois différentiable, alors $\nabla^2 f(x^*) = [\partial^2 / \partial x_i \partial x_j (x^*)]$ est une matrice semi-définie positive.

La condition (a) est appelée **condition nécessaire du premier ordre**, en ajoutant la condition (b) on a la **condition nécessaire du second ordre**.

Preuve. Soit x^* un minimum local (ou global) de f .

Comme f est deux fois continûment différentiable, le développement de Taylor à l'ordre 2 au voisinage de x^* donne:

$$f(x) = f(x^*) + \nabla f(x^*)^T (x - x^*) + \frac{1}{2} (x - x^*)^T \nabla^2 f(x^*) (x - x^*) + \|x - x^*\|^2 \theta (x - x^*)$$

avec $\theta(x - x^*) \rightarrow 0$ quand $x \rightarrow x^*$

1. Si $\nabla f(x^*) \neq 0$ alors en choisissant $x = x^* - \theta \nabla f(x^*)$ on aurait, pour $\theta > 0$ suffisamment petit: $f(x) < f(x^*)$ ce qui contredirait le fait que x^* est un minimum local. donc la condition (a) est bien nécessaire.

2. Si la matrice $\nabla^2 f(x)$ n'est pas semi-définie positive ,c'est qu'il existe un vecteur $d \in \mathbb{R}^n (d \neq 0)$ tel que: $d^T \nabla^2 f(x^*) d < 0$. En choisissant alors $x = x^* + \theta d$,pour $\theta > 0$ suffisamment petit on aurait $f(x) > f(x^*)$ ce qui contredirait encore l'optimalité locale de x^* .

Remarque 2.1.1. En pratique,la condition nécessaire du second ordre est difficile à vérifier systématiquement, car elle exige de calculer les dérivées secondes et d'analyser les valeurs propres de la matrice hessienne.

La condition nécessaire d'optimalité du premier ordre joue un rôle central en optimisation.les vecteurs x qui vérifient cette condition sont appelés des points critiques ou points stationnaires.parmis eux ,il y'a des minima locaux ,des maxima locaux et des points qui ne sont ni l'un ni l'autre.ces derniers sont appelés points selle.

Définition 2.1.1. "Point critique"

Soit $f: \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Tout vecteur $x \in \mathbb{R}^n$ qui vérifie la condition (a) c'est-à-dire: $\nabla f(x) = 0$ est appelé un point stationnaire ou point critique de f .

2.1.3 Conditions suffisantes d'optimalité locale

Théorème 2.1.2 (2). "Conditions suffisantes d'optimalité locale"

Soit $f: \mathbb{R}^n \rightarrow \mathbb{R}$. Une condition suffisante pour que x^* soit un optimum local de f est:

1. $\nabla f(x^*) = 0$ (stationnarité)
2. La hessienne $\nabla^2 f(x^*)$ est une matrice définie positive

Preuve.

Considérons un point x^* satisfaisant les deux conditions (1) et (2) du théorème.

Le développement de Taylor de f au voisinage de x^* s'écrit alors :

$$f(x) = f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*) + \|x - x^*\|^2 \theta(x - x^*)$$

Avec $\theta(x - x^*) \rightarrow 0$ quand $x \rightarrow x^*$

Pour toute direction de déplacement $d \in \mathbb{R}^n (\|d\| = 1)$ on a :

$$f(x^* + td) = f(x^*) + \frac{t^2}{2} d^T \nabla^2 f(x^*) d + t^2 \theta(t)$$

d'où

$$f(x^* + td) - f(x^*) = \frac{t^2}{2} d^T \nabla^2 f(x^*) d + t^2 \theta(t)$$

où $\theta(t) \rightarrow 0$ quand $t \rightarrow 0$.

En vertu de la condition (2) on a:

$$\frac{t^2}{2} d^T \nabla^2 f(x^*) d > 0$$

et par suite, pour θ suffisamment petit, on a

$$f(x^* + td) > f(x^*)$$

Ce qui montre que x^* est bien un minimum local de f .

Nous présentons maintenant une condition suffisante pour qu'un minimum local soit également un minimum global.

Conditions suffisantes d'optimalité globale

Dans le cas d'une fonction convexe f définie sur \mathbb{R}^n , on a la condition nécessaire et suffisante pour que x^* soit un minimum global de f est donnée dans le théorème suivant:

Théorème 2.1.3. [3] "Conditions suffisantes d'optimalité globale"

Soit une fonction continue $f: \mathbb{R}^n \rightarrow \mathbb{R}$ et $x^ \in \mathbb{R}^n$ un minimum local de f .*

- *Si f est une fonction convexe, alors x^* est un minimum global de f .*
- *Si de plus f est strictement convexe, x^* est l'unique minimum global de f .*

Autrement dit, dans le cas convexe, la stationnarité à elle seule constitue une condition nécessaire et suffisante d'optimalité globale.

Preuve.

Supposons par l'absurde qu'il existe un autre minimum local $x^+ \neq x^*$, tel que $f(x^+) < f(x^*)$

Par la convexité de f , nous avons

$$f(\lambda x^* + (1 - \lambda)x^+) \leq \lambda f(x^*) + (1 - \lambda)f(x^+),$$

où $0 \leq \lambda \leq 1$ Comme $f(x^+) < f(x^*)$, nous avons pour chaque $\lambda \in]0,1[$

$$f(\lambda x^* + (1 - \lambda)x^+) < \lambda f(x^*) + (1 - \lambda)f(x^*) = f(x^*), \quad (2.2)$$

Considérons $\varepsilon > 0$ arbitraire, et montrons que la définition d'un minimum local est contredite.

Si $\varepsilon \geq \|x^* - x^+\|$, la définition d'un minimum local n'est pas vérifiée pour $x = x^+$, en prenant $\lambda = 1$ dans (2.2).

Si $\varepsilon < \|x^* - x^+\|$, considérons $0 < \eta < 1$ tel que $\varepsilon < \|\eta x^* + (1 - \eta)x^+\| = \varepsilon$. Dans ce cas, la définition d'un minimum local n'est pas vérifiée pour $x = (\lambda x^* + (1 - \lambda)x^+)$ avec

$\eta \leq \lambda < 1$ par (2.2). Comme $\eta < 1$, de tels λ existent toujours.

Considérons maintenant une fonction strictement convexe, et supposons que x^* et y^* soient deux minima globaux distincts, et donc $x^* \neq y^*$ et $f(x^*) = f(y^*)$. En fonction de la définition de convexité stricte, nous avons

$$f(\lambda x^* + (1 - \lambda)y^*) < \lambda f(x^*) + (1 - \lambda)f(y^*) = f(x^*) = f(y^*),$$

ce qui contredit que x^* et y^* sont des minima globaux.

Nous terminons cette section par une discussion des conditions d'optimalité pour les problèmes quadratiques.

2.1.4 Conditions d'optimalité pour les problèmes quadratiques

Parmi les fonctions non linéaires, les fonctions quadratiques joueront un rôle important dans les algorithmes d'optimisation.

Définition 2.1.2. "Fonction quadratique"

Une fonction $f: \mathbb{R}^n \rightarrow \mathbb{R}$ sera dite quadratique si elle peut s'écrire

$$f(x) = \frac{1}{2}x^T A x + b^T x + c \quad (2.3)$$

où A est une matrice symétrique $n \times n$, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$. Nous avons alors

$$\nabla f(x) = A x + b \quad (2.4)$$

et

$$\nabla^2 f(x) = A \quad (2.5)$$

Théorème 2.1.4. [3] "Conditions d'optimalité pour les problèmes quadratiques"

Considérons le problème

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}x^T A x + b^T x + c \quad (2.6)$$

où A est une matrice symétrique $n \times n$, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$.

1. Si A n'est pas semi-définie positive, alors le problème (2.6) ne possède pas de solution, c'est-à-dire qu'il n'existe aucun $x \in \mathbb{R}^n$ qui soit un minimum local de (2.3).
2. Si A est définie positive, alors

$$x^* = -A^{-1}b \quad (2.7)$$

est l'unique minimum global de (2.3)

Preuve.

Nous avons $\nabla f(x) = Ax + b$ et $\nabla^2 f(x) = A$

1. Supposons par l'absurde qu'il existe x^* minimum local de (2.3). D'après (b) du théorème (2.1.1), $\nabla^2 f(x) = A$ est semi-définie positive, ce qui contredit l'hypothèse.
2. Comme A est définie positive, le point x^* dans (2.7) est bien défini car:

$$\nabla f(x^*) = Ax^* + b$$

On remplace x^* par sa valeur, on aura

$$\nabla f(x^*) = -AA^{-1}b + b = 0$$

Les conditions suffisantes d'optimalité du théorème (2.1.2) sont vérifiées et x^* est un minimum local de f . De plus, f est convexe. D'après le théorème (2.1.3), x^* est l'unique minimum global.

2.1.5 Méthodes itératives

On considère le problème d'optimisation sans contrainte suivant :

$$(P) \begin{cases} \text{Min } f(x) \\ x \in \mathbb{R}^n \end{cases}$$

Pour résoudre le problème (P) intéressons nous aux méthodes suivantes:

1. *Les méthodes basées sur le gradient*
2. *Les méthodes de Newton*
3. *La méthode des directions conjuguées*

Avant de citer ces méthodes ,nous allons rappeler brièvement les définitions de quelques notions importantes pour la suite.

Notion de direction de descente

Le gradient joue un rôle essentiel en optimisation. Dans le cadre des méthodes d'optimisation, il sera également important d'analyser le comportement de la fonction objectif dans certaines directions. Commençons pour cela par rappeler le concept de dérivée directionnelle.

Définition 2.1.3. "*dérivée directionnelle*"

Soit $f : \mathbb{R}^n \mapsto \mathbb{R}$ une fonction continue, soit $x \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$.
La dérivée directionnelle de f en x dans la direction d est définie par :

$$df(x,d) = \lim_{t \leftarrow 0^+} \frac{f(x+td) - f(x)}{t} \quad (2.8)$$

si cette limite existe.

Proposition 2.1.

Si f est différentiable en un point $x \in \mathbb{R}^n$, alors pour tout $d \neq 0$, f admet une dérivée dans la direction d en x et :

$$df(x,d) = d^T \nabla f(x) \quad (2.9)$$

On rappelle que la réciproque est fautive! La dérivabilité selon tout vecteur en x n'implique pas nécessairement la différentiabilité de f en x .

La dérivée directionnelle donne des informations sur la pente de la fonction dans la direction d , tout comme la dérivée donne des informations sur la pente des fonctions à une variable. En particulier,

- si $df(x,d) > 0$ alors f est croissante dans la direction d .
- si $df(x,d) < 0$ alors f est décroissante dans la direction d .

Dans ce dernier cas, on dira que d est une direction de descente de f .

Définition 2.1.4. "direction de descente"

Soient $f : \mathbb{R}^n \mapsto \mathbb{R}$, une fonction différentiable et $x \in \mathbb{R}^n$. Le vecteur $d \in \mathbb{R}^n$ est une direction de descente pour f à partir du point x si $t \mapsto f(x+td)$ est décroissante en $t = 0$, c'est-à-dire s'il existe $\eta > 0$ tel que :

$$\forall t \in]0,\eta], \quad f(x+td) < f(x) \quad (2.10)$$

Théorème 2.1. [9]"Direction de plus forte descente"

Soit $f : \mathbb{R}^n \mapsto \mathbb{R}$ une fonction fonction différentiable, et x un point de \mathbb{R}^n . Alors pour toute direction d de norme constante égale à $\|d\| = \|\nabla f(x)\|$, on a :

$$(-\nabla f(x))^T \nabla f(x) \leq d^T \nabla f(x)$$

La direction $d^* = -\nabla f(x)$ est appelée direction de plus forte descente.

La direction opposée au gradient est donc celle dans laquelle la fonction a la plus forte descente, le gradient correspond, lui, à la plus forte pente.

Corollaire 2.1. [9]"Direction de plus forte pente"

Le vecteur $\nabla f(x)$ est appelée direction de plus forte pente de f au point $x \in \mathbb{R}^n$

On remarquera que si x^* est un point de minimum local de f , alors il n'existe aucune direction de descente pour f au point x^* .

Proposition 2.2.

Soient $f: \mathbb{R}^n \mapsto \mathbb{R}$ différentiable et $x \in \mathbb{R}^n$ tel que : $\nabla f(x) \neq 0$.

Le vecteur $d \in \mathbb{R}^n$ est une direction de descente pour f à partir du point x ssi la dérivée directionnelle de f en x dans la direction d vérifie :

$$df(x,d) = \nabla f(x)d < 0 \quad (2.11)$$

Théorème 2.2. [9] " descente "

Soit $f: \mathbb{R}^n \mapsto \mathbb{R}$ une fonction différentiable. Soient $x \in \mathbb{R}^n$ tel que $\nabla f(x) \neq 0$ et $d \in \mathbb{R}^n$. Si d est une direction de descente en x alors il existe η tel que :

$$\forall \alpha \in [0,\eta], \quad f(x + \alpha d) < f(x) \quad (2.12)$$

En utilisant ce résultat, on construit un algorithme général de minimisation, nommée algorithme de descente. Il consiste simplement à suivre une direction de descente de façon itérative jusqu'à l'obtention d'un "bon" minimiseur.

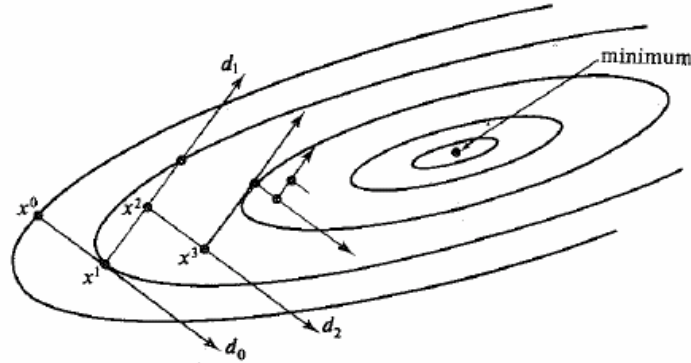
Déduction d'une classe d'algorithmes de descente

Dans cette section, nous construisons une famille d'algorithmes itératifs pour calculer un minimum local d'une fonction f . La base de notre étude est la suivante: si un point x ne satisfait pas aux conditions d'optimalité, il est possible de produire un autre point, x^1 pour lequel $f(x^1) < f(x^0)$. Cette famille d'algorithmes porte le nom de méthodes de descente. On cherche alors x^1 sous la forme $x^1 = x^0 + \alpha_1 d^1$ où d^1 est un vecteur non nul de \mathbb{R}^n et α_1 un réel strictement positif. En pratique donc, on cherche d^1 et α^1 pour que $f(x^0 + \alpha_1 d^1) < f(x^0)$. On ne peut pas toujours trouver d^1 .

Quand d^1 existe on dit que c'est une direction de descente et α_1 est le pas de descente. La direction et le pas de descente peuvent être fixes ou change à chaque itération. Le schéma général d'une méthode de descente est le suivant :

$$\begin{cases} x^0 \in \mathbb{R}^n \\ x^{k+1} = x^k + \alpha_k d^k, \quad d^k \in \mathbb{R}^n - \{0\}, \quad \alpha_k \in \mathbb{R}^{+*} \end{cases}$$

où α_k et d^k sont choisis de telle sorte que $f(x^0 + \alpha_1 d^1) < f(x^0)$.

FIG. 2.1 – *Schéma général d'une méthode de descente***Algorithme de descente**1. **Initialisation**

$k \leftarrow 0$: On se donne une fonction f différentiable, un point initial x^0 et un seuil de tolérance $\varepsilon > 0$

2. **Itération** $k = 1, 2, \dots$

$x^{k+1} \leftarrow x^k + \alpha_k d^k$

3. **Critère d'arrêt**

Si $\|x^{k+1} - x^k\| < \varepsilon$ **STOP** x^k est la solution optimale

Sinon, on pose $k \leftarrow k + 1$ et on retourne à 2.

Il est important de noter que l'algorithme n'est pas complètement défini. En effet, rien n'est spécifié pour le pas α , et la direction d . Enfin, certaines hypothèses supplémentaires sont nécessaires afin de garantir que la méthode converge.

2.1.6 La recherche linéaire d'un pas

Soit x un point de \mathbb{R}^n tel que: $\nabla f(x) \neq 0$ et d une direction de descente de f en x . Intéressons nous maintenant d'un peu plus près à la phase de recherche linéaire i.e. au calcul d'un pas $\alpha \in \mathbb{R}_+^*$ vérifiant:

$$f(x + \alpha d) < f(x).$$

D'après la caractérisation de la descente (proposition 2.2), il s'agit donc à chaque itération k , de trouver un point x^{k+1} dans une direction d vérifiant :

$$\varphi'(0) = \nabla f(x)^T d < 0$$

Méthode du gradient à pas optimal

La méthode de plus profonde descente est la méthode de gradient la plus utilisée. Une idée naturelle consiste à suivre la direction de plus forte descente et à faire un pas qui rend la fonction à minimiser la plus petite possible dans cette direction. La première question est de choisir cette direction. On remplace f au voisinage de x^k par son développement de Taylor de premier ordre:

$$f(x^k + d) \simeq f(x^k) + \nabla f(x^k)^T d$$

On voudrait que la dérivée directionnelle $\nabla f(x^k)^T d$ soit la plus petite possible dans un voisinage de d . On cherche donc à résoudre :

$$\min_{d \in \mathbb{R}^n} \nabla f(x^k)^T d \quad s.c \quad \|d\| = 1$$

dont la solution nous ait donnée par le théorème de la direction de plus forte descente, comme la direction de plus profonde descente normalisée :

$$d^k = -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$$

On pose ensuite $q(\alpha) = f(x^k - \alpha \nabla f(x^k))$ et on calcule α_k de façon à minimiser q pour $\alpha \geq 0$. On est alors ramené à un problème d'optimisation unidimensionnelle.

Algorithme de plus profonde descente ("Steepest descent")

1. *Initialisation*

$k \leftarrow 0$: On se donne une fonction f différentiable, un point initial x^0 et un seuil de tolérance $\varepsilon > 0$

2. *Itération* $k = 1, 2, \dots$

pose $d^k = -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$

calculer: $\alpha_k = \arg \min_{\alpha > 0} f(x^k + \alpha d^k)$

$x^{k+1} \leftarrow x^k + \alpha_k d^k$

3. *Critère d'arrêt*

Si $\|x^{k+1} - x^k\| < \varepsilon$ **STOP**) x^{k+1} est la solution optimale

Sinon, on pose $k \leftarrow k + 1$ et on retourne à 2.

Convergence de la méthode du gradient

On veut minimiser une fonction f . Pour cela on se donne un point de départ arbitraire x^0 . Pour construire l'itéré suivant x^1 il faut penser qu'on veut se rapprocher du minimum de f , on veut donc que:

$$f(x^1) < f(x^0)$$

Notre but est maintenant de produire une suite de points x^k dont les points d'accumulation sont stationnaires, et ce quel que soit le point de départ x^0 de la suite. Nous utiliserons pour ce faire le fait que nous tentons de minimiser une fonction f , ce qui nous aide à évaluer la qualité relative de deux points consécutifs de la suite.

Remarque 2.1.2.

Le gradient à pas optimal est en principe la meilleure méthode des méthodes du gradient d'un point de vue de la rapidité de convergence.

La Méthode de Newton

En réalité la méthode de Newton n'est pas une méthode d'optimisation, elle est utilisée pour résoudre des équations non linéaires de la forme $f(x) = 0$ ou f est une fonction de \mathbb{R}^n dans \mathbb{R}^n . Nous allons commencer par la décrire puis on montre comment l'appliquer à la recherche de minimum

La méthode de Newton dans \mathbb{R}

Présentons d'abord formellement cette méthode dans \mathbb{R} . Soit \mathbb{I} un intervalle de \mathbb{R} et $f : \mathbb{I} \rightarrow \mathbb{R}$, pour résoudre $f(t) = 0$ où f est une fonction C^1 . La méthode de Newton part d'une solution approchée x et remplace l'équation $f(t) = 0$ par l'équation approchée $f(x) + (t - x)f'(x) = 0$, d'où la solution

$$t = x - \frac{f(x)}{f'(x)} \quad (2.13)$$

Bien entendu, cette formule n'a un sens que si $f'(x) \neq 0$. On espère que le nombre réel t donné par cette équation est un peu plus proche d'une racine que ne l'était x . En tout cas, si t appartient encore à \mathbb{I} , on peut recommencer en partant de t , etc..., d'où une suite définie (pas forcément pour tout n) par la relation de récurrence :

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)} \quad (2.14)$$

Nous préciserons ensuite les conditions d'utilisation de la méthode et justifierons l'existence des itérés successifs dans un théorème général de convergence.

Algorithme de Newton dans \mathbb{R}

1. *Initialisation*

$k = 0$: On se donne une fonction f différentiable, un point initial x^0 et un seuil de tolérance $\varepsilon > 0$

2. *Itération* $k=1,2,\dots$

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)};$$

3. *Critère d'arrêt*

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP) x^{k+1} est la solution optimale

Si non, on pose $k = k + 1$ et on retourne à 2.

Remarque 2.1.3.

Remarquons qu'il faut non seulement assurer la convergence de la suite x^k vers la solution x^* , mais aussi montrer que cette suite est bien définie, c'est-à-dire montrer que $f'(x^k) \neq 0$ à l'étape 2. Cette méthode est aussi appelée *méthode de la tangente*. En effet chaque itéré x^{k+1} est obtenu à partir du précédent en traçant la tangente à la courbe de f au point $(x^k, f(x^k))$ et en prenant son intersection avec l'axe des abscisses (Fig 2.1).

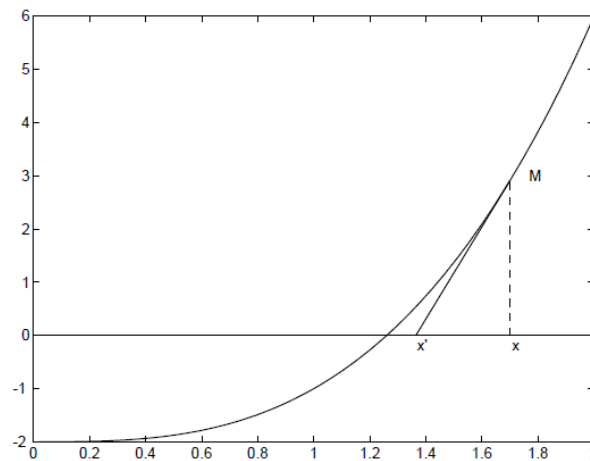


FIG. 2.2 – *Interprétation géométrique de la méthode de Newton*

La méthode de Newton dans \mathbb{R}^n

Nous pouvons maintenant généraliser à \mathbb{R}^n . Soit f une fonction de classe \mathcal{C}^1 de \mathbb{R}^n à valeurs dans \mathbb{R}^n . On suppose que l'équation

$$f(x) = 0$$

On suppose ici que f est deux fois continûment dérivable et que l'on sait calculer ses dérivées secondes. Au voisinage d'un point x^k , on approche f par la fonction quadratique donnée par la formule de Taylor d'ordre 2 :

$$q(x) = f(x^k) + (x - x^k)^t \cdot \nabla f(x^k) + \frac{1}{2} (x - x^k)^t \cdot \nabla^2 f(x^k) \cdot (x - x^k) \quad (2.15)$$

On peut alors choisir pour x^k le point, s'il existe, qui minimise q ; pour que ce point minimisant q existe, il est suffisant que $\nabla^2 f(x^k)$ soit définie positive, il est alors déterminé

par l'équation $\nabla q(x) = 0$ qui s'écrit :

$$\nabla f(x^k) + \nabla^2 f(x^k) \cdot (x - x^k) = 0$$

d'où :

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k) \quad (2.16)$$

On pose

$$d^k = -[\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k) \quad (2.17)$$

On a $\nabla^2 f(x^k)$ est définie positive, la direction d^k sera le minimum de (2,11).

Dans ce cas,

$$\nabla f(x^k) d^k = -(d^k)^t \nabla^2 f(x^k) d^k \leq 0 \quad (2.18)$$

on obtient bien une direction de descente.

Définition 2.1.5. "Direction de Newton"

Soit à minimiser la fonction f , \mathcal{C}^2 sur son domaine de définition, alors on désigne par direction de Newton en x le vecteur

$$d = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

Algorithme de Newton dans \mathbb{R}^n

1. Initialisation

$k = 0$: On se donne une fonction f différentiable, un point initial x^0 et un seuil de tolérance $\varepsilon > 0$

2. Itération $k = 1, 2, \dots$

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k);$$

3. Critère d'arrêt

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP) x^{k+1} est la solution optimale

Si non, on pose $k = k + 1$ et on retourne à 2.

Convergence de la méthode de Newton

L'inconvénient majeur de la méthode est sa sensibilité au choix du point de départ: la convergence est *locale*. Si ce point est mal choisi ("trop loin" de la solution) la méthode peut diverger. Un autre inconvénient de la méthode de Newton réside également dans le fait que nous avons à évaluer et "à inverser" $\nabla^2 f(x^k)$ à chaque itération. Certaines méthodes proposent d'utiliser non pas $\nabla^2 f(x^k)$ comme matrice mais plutôt une approximation de celle-ci (plus précisément, dans les méthodes de Quasi-Newton)^[14]. L'avantage de cette méthode est sa grande rapidité. La convergence est quadratique, c'est-à-dire que l'erreur

$e_k = \|x^{k+1} - x^k\|$ est élevée au carré à chaque itération. Concrètement, si elle vaut 10^{-2} à l'étape k elle vaudra 10^{-4} à l'étape $k + 1$ et 10^{-8} à l'étape $k + 2$.

Proposition 2.1.1.

Si x^0 est choisi suffisamment proche d'un minimum local x^ où le hessien de f est défini positif, alors la suite (x^k) a une convergence quadratique vers x^* .*

La méthode du gradient conjugué

L'algorithme que nous présentons dans cette section possède deux intérêts. Il permet de résoudre des systèmes linéaires à coefficients strictement positifs de grande taille et il sert d'algorithme de base pour la résolution des problèmes d'optimisation non linéaire. Soit le problème de minimisation quadratique suivant:

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c \quad \text{avec } x \in \mathbb{R}^n \quad (2.19)$$

où A est une matrice carrée d'ordre n , symétrique, définie positive, et $b \in \mathbb{R}^n$.

On note

$$\nabla f(x) = Ax + b \quad (2.20)$$

qui est non nul, et

$$\nabla^2 f(x) = A \quad (2.21)$$

La première idée fondamentale de l'algorithme du gradient conjugué consiste à choisir chaque direction de descente conjuguée à la direction de descente précédente par rapport à A . Afin de développer le gradient conjugué, introduisons la notion de direction conjuguée.

Définition 2.1.6. "directions conjuguées"

Soit une matrice $A \in \mathbb{R}^n \times \mathbb{R}^n$ définie positive. Les vecteurs (ou directions) non nuls de \mathbb{R}^n d^1, \dots, d^n sont conjugués par rapport à A (A -conjugués) si

$$(d^i)^T A d^j = 0, \quad \forall i, j \text{ tels que } i \neq j \quad (2.22)$$

Ceci signifie que ces deux vecteurs sont orthogonaux pour le produit scalaire associé à la matrice A , défini par

$$\langle x, y \rangle_A = x^T A y, \quad \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \quad (2.23)$$

Théorème 2.1.5. *Soit $\{d^1, \dots, d^k\}$ un ensemble de directions non nulles et conjugués par rapport à A . Alors les vecteurs d^1, \dots, d^n forme une base de \mathbb{R}^n .*

Preuve.

On a n vecteurs dans \mathbb{R}^n pour montrer qu'ils forment une base de \mathbb{R}^n , il suffit de montrer qu'ils sont linéairement indépendants

Soit $\{d^1, \dots, d^n\}$ est une famille de n directions A-conjuguées, et $\lambda_i \in \mathbb{R}$ avec $(i = 1, \dots, n)$.

Montrer que

$$\sum_{i=1}^{i=n} \lambda_i d^i = 0 \implies \lambda_i = 0$$

On multiplie dans les deux membres par Ad^j avec $j = 1, \dots, n$

$$\sum_{i=1}^{i=n} \lambda_i d^i Ad^j = \lambda_1 (d^1)^T Ad^j + \lambda_2 (d^2)^T Ad^j + \dots + \lambda_k (d^k)^T Ad^j = 0 \quad \text{avec } j = 1, \dots, n$$

Comme d^1, \dots, d^n sont des directions A-conjuguées alors

$$(d^1)^T Ad^j = (d^2)^T Ad^j = \dots = (d^i)^T Ad^j = 0$$

D'où

$$\lambda_j = 0 \quad \text{avec } j = 1, \dots, n$$

Donc

d^1, \dots, d^n sont linéairement indépendants $\implies d^1, \dots, d^n$ forment une base de \mathbb{R}^n

Calcul des directions conjuguées

On peut construire une base de vecteurs propres orthogonaux car A est symétrique. Ces vecteurs sont par construction A-conjugués et constituent un choix possible de directions conjuguées. Mais le calculs de ces vecteurs est en général très coûteux en temps de calculs. En fait on peut calculer les directions conjuguées au fur et à mesure qu'on en a besoin par la récurrence suivante

$$d^k = -\nabla f(x^k) + \beta^k d^{k-1}$$

qui s'interprète comme l'opposé du gradient au nouveau point altéré par la direction précédente. Pour que d^n et d^{n-1} soient des directions A-conjuguées on choisit

$$\beta^n = \frac{(\nabla f(x^n))^T \nabla f(x^n)}{(\nabla f(x^{n-1}))^T \nabla f(x^{n-1})} \quad (2.24)$$

La direction d^n est également A-conjuguée avec les directions d^i pour $i = 0, \dots, n-2$.

Algorithme du gradient conjugué**1. Initialisation**

fixer $\varepsilon > 0$, choisir $x^0 \in \mathbb{R}^n$ poser $\nabla f(x^0) = Ax^0 - b$ et $d^0 = -\nabla f(x^0)$

2. **Itération** $k = 0, 1, \dots$

$$\text{calculer } \alpha_k = -\frac{(d^k)^T \nabla f(x^k)}{(d^k)^T A d^k}$$

$$\text{calculer } x^{k+1} = x^k + \alpha_k d^k$$

$$\text{calculer } \nabla f(x^{k+1}) = \nabla f(x^k) + \alpha_k A d^k$$

$$\text{calculer } \beta^{k+1} = \frac{\|\nabla f(x^{k+1})\|^2}{\|\nabla f(x^k)\|^2}$$

$$\text{calculer } (d^{k+1}) = -\nabla f(x^{k+1}) + \beta^{k+1} d^k$$

3. **Critère d'arrêt**

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP) x^{k+1} est la solution optimale

Si non, on pose $k = k + 1$ et on retourne à 2.

Convergence de la méthode du gradient conjugué

Concernant la convergence de la méthode du gradient conjugué, on a le résultat suivant:

Théorème 2.1.6. [18]

Soit A une matrice symétrique définie positive d'ordre n . Toute méthode qui utilise des directions conjuguées conduit à la solution exacte en au plus n itérations.

Preuve.

Les directions d^0, d^1, \dots, d^{n-1} forment une base A -orthogonale de \mathbb{R}^n . De plus, puisque x^k est optimal par rapport à toutes les directions d^j , ($j = 0, \dots, k-1$), le vecteur r^k est orthogonal à l'espace $V^{k-1} = \text{vect}(d^0, d^1, \dots, d^{k-1})$. Par conséquent,

$$r^n \perp V^{n-1} = \mathbb{R}^n$$

et donc $r^n = 0$ ce qui implique

$$x^n = x$$

2.2 Optimisation sous contraintes

Dans de nombreuses applications pratiques, les variables d'une fonction donnée sont soumises à certaines conditions ou contraintes. Ces contraintes peuvent être formulées sous la forme d'égalité ou d'inégalité. D'où la modélisation suivante

$$(p) \left\{ \begin{array}{l} \text{Min}f(x) \\ \text{sous les contraintes} \\ g_i(x) \leq 0 \quad i = \{1,2,\dots,m\} \\ h_j(x) = 0 \quad j = \{1,2,\dots,k\} \\ x \in S \end{array} \right.$$

L'ensemble $S = \{x \in \mathbb{R}^n / g_i(x) \leq 0 \quad i = 1,2,\dots,m \quad h_j(x) = 0 \quad j = 1,2,\dots,k\}$ est appelé le domaine des contraintes. Tout vecteurs $x \in \mathbf{R}^n$ vérifiant: $x \in S$ est dit une solution admissible(réalisable) du problème (p).

Une difficulté importante en optimisation avec contraintes consiste à savoir déplacer dans l'ensemble des contraintes,i.e étant donnée comment garantir que la direction de recherche reste toujours dans l'ensembles des contraintes.

2.2.1 Conditions nécessaires d'optimalités

L'énoncé des conditions d'optimalité nécessite l'introduction de multiplicateurs de Lagrange. Dans un premier temps, on va s'intéresser à un problème sous contraintes simplifié, dans lequel ne figurent que des contraintes égalités.

Multiplicateurs de Lagrange, le théorème des extrema liés

On cherche donc à résoudre

$$(p) \left\{ \begin{array}{l} \text{Min}f(x) \\ g_i(x) = 0 \quad i = \{1,2,\dots,m\} \\ x \in \mathbb{R}^n \end{array} \right.$$

La méthode des multiplicateurs de Lagrange consiste à construire une fonction auxiliaire

$L(x, \lambda)$, appelée Lagrangien, définie ainsi : $L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x)$ (2.27)

Définition 2.2.1. Soit $G(x) = (g_1(x), g_2(x), \dots, g_m(x))$ la matrice jacobienne de G est : $JG(x) = (\nabla g_1(x), \dots, \nabla g_m(x))^T_{m \times n}$
 On dit que $JG(x)$ est de rang plein si $rg JG(x) = m$

Définition 2.2.2. "Le point régulier"

on dit qu'un point x^* est régulier si $rg JG(x^*) = m$.

On dit aussi qu'il vérifie la condition de qualification des contraintes.

Théorème 2.2.1. "Condition nécessaire de lagrange

Soit x^* un minimum local régulier pour le problème (p)

alors il existe $\lambda_1^*, \dots, \lambda_m^* \in \mathbb{R}$ tq $\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) = 0$

c-à-d $\partial f(x^*)/\partial x_k + \sum_{i=1}^m \lambda_i \partial g_i(x^*)/\partial x_k = 0 \quad k = 1, \dots, n$

Remarque 2.2.1. La méthode des multiplicateurs de Lagrange ne peut pas être utilisée dans tous les cas. Notamment lorsque le problème d'optimisation possède des contraintes d'inégalités

Définition 2.2.3. "Directions admissibles "

Soit un problème générale d'optimisation (P), et soit $x \in \mathbb{R}^n$ admissible. Une direction d est dite admissible en x s'il existe un $\eta > 0$ tel que $x + \alpha d$ est admissible pour tout $0 < \alpha \leq \eta$.

Définition 2.2.4. " Les indices actifs"

Soit l'ensembles des contraintes $S = \{x \in \mathbb{R}^n / g_i(x) \leq 0, i \in \mathbb{I} = \{1, 2, \dots, m\}\}$.

L'ensemble des indices actifs (l'ensemble des indices des contraintes saturées) au point x^* est formé des indices i qui vérifient $g_i(x^*) = 0$. Il est noter comme suit:

$$\mathbb{I}_a = \mathbb{I}_a(x^*) = \{i \in \mathbb{I} / g_i(x^*) = 0\}$$

Définition 2.2.5. "Qualification des contraintes (Q.c)"

Soit un problème d'optimisation suivant:

$$(p) \begin{cases} \text{Min} f(x) \\ g_i(x) \leq 0 \quad i = \{1, 2, \dots, m\} \\ x \in \mathbb{R}^n \end{cases}$$

soit un point admissible x^* . La condition de qualification des contraintes (Kuhn et Tucker

1961, Abadie 1967)^[4] est vérifiée si tout élément du cône des directions en x^* est une direction admissible à la limite en x^* .

Donc (Q.c) en x^* , revient à chercher l'existence d'une direction d pointant à l'intérieur du cône défini par :

$$K = \{d \in \mathbb{R}^n / \nabla^T g_i(x^*)d \leq 0, \quad \forall i \in \mathbb{I}_a(x^*)\}. \quad (2.29)$$

C'est à dire que :

$$(Q.c) \iff \exists d \in \mathbb{R}^n / \nabla^T g_i(x^*)d \leq 0, \quad i \in \mathbb{I}_a(x^*). \quad (2.30)$$

Evidemment, la vérification directe de (Q.c) peut être difficile en pratique, et c'est pourquoi on a recherché des conditions suffisantes pour que (Q.c) soit réalisée. Les résultats les plus importants sont rassemblés dans le lemme ci-dessous.

Lemme 2.2.1. [3]

Pour que (Q.c) soit vérifiée en tout point $x \in X$ il suffit que l'une des conditions (a) ou (b) soit réalisée :

- a) toutes les fonctions g_i , sont linéaires (Karlin 1959)^[5],
- b) toutes les fonctions g_i sont convexes et X a un intérieur non vide (Slater 1950)^[6].

Pour que (Q.c) soit vérifiée en un point $x^* \in X$, il suffit que l'on ait :

- c) les gradients $\nabla g_i(x^0)$ ($i \in \mathbb{I}^*$) des contraintes saturées en x^* sont linéairement indépendants (Fiacco et McCormick 1968)^[7].

2.2.2 Conditions nécessaires Karush-Kuhn et Tucker

Les conditions nécessaires de Karush-Kuhn et Tucker (K.K.T)^[8] donnée dans le théorème suivant est fondamentale. Sous l'hypothèse de qualification des contraintes, une condition nécessaire d'optimalité locale pour un problème d'optimisation avec contraintes du type

$$\left\{ \begin{array}{l} \min f(x) \\ \text{Sous les contraintes} \\ g_i(x) \leq 0 \quad i \in \mathbb{I} \\ x \in \mathbb{R}^n \end{array} \right. \quad (2.31)$$

Théorème 2.2.2. [8] "Karush-Kuhn et Tucker 1951"

On suppose que les fonctions f et g_i ($i \in \mathbb{I}$) sont continûment différentiables, et que l'hypothèse de qualification des contraintes est vérifiée en $x^* \in X$, avec :

$$X = \{x \in \mathbb{R}^n / g_i(x) \leq 0, \quad \forall i \in \mathbb{I}\}.$$

Alors, une condition nécessaire pour que x^* soit un optimum local de (2.31) est qu'il existe des nombres $\lambda_i \geq 0 (i \in \mathbb{I})$ appelés multiplicateurs de Karush-Kuhn et Tucker, tels que :

$$\begin{cases} \nabla f(x^*) + \sum_{i \in \mathbb{I}} \lambda_i \nabla g_i(x^*) = 0 & , \\ \text{et,} \\ \lambda_i \nabla g_i(x^*) = 0 & (i \in \mathbb{I}) \end{cases} \quad (2.32)$$

Pour démontrer ce théorème, nous aurons besoin des deux résultats suivants :

Lemme 2.2.2. [3]”Farkas”

Soit A une matrice $n \times m$, b un vecteur de \mathbb{R}^m , avec $m < n$.

Si pour chaque vecteur x vérifiant $Ax \leq 0$, et $cx \leq 0$ alors il existe des coefficients $\lambda_i \geq 0$, tels que

$$c = \sum_{i=1}^m \lambda_i A_i$$

Lemme 2.2.3. [3]

Soit d une direction admissible en x^* . Alors nécessairement, d vérifie les relations :

$$\nabla g_i^T(x^*).d \leq 0 \quad (\forall i \in \mathbb{I}_a). \quad (2.33)$$

Preuve. ”Théorème de Karush-Kuhn-Tucker”

Le point x^* étant un minimum, alors pour toute direction admissible d en x^* , on aura $\nabla f(x^*)^T.d \geq 0$, et ce, en vertu du lemme (2.2.3). En utilisant (2.33), on peut donc écrire

$$\nabla g_i^T(x^*).d \leq 0 \quad (\forall i \in \mathbb{I}_a) \implies -\nabla f(x^*)^T.d \leq 0$$

En vertu du lemme de Ferkas (2.2.2), il existe alors des coefficients λ_i^* , ($i \in \mathbb{I}_a$), tels que

$$-\nabla f(x^*) = \sum_{i \in \mathbb{I}_a} \lambda_i^* A_i$$

Posons $\lambda_i^* = 0$ pour $i \in \mathbb{I} \setminus \mathbb{I}_a(x^*)$. Donc on obtient

$$\begin{cases} -\nabla f(x^*) = \sum_{i \in \mathbb{I}} \lambda_i^* A_i \iff \nabla f(x^*) + \sum_{i \in \mathbb{I}} \lambda_i^* \nabla g_i(x^*) = 0 & , \\ \text{et,} \\ \lambda_i^* \nabla g_i(x^*) = 0 & (i \in \mathbb{I}) \end{cases} \quad .$$

Remarque 2.2.2.

Dans le cas où les gradients des contraintes saturées sont linéairement indépendants (x^* est alors appelé un point régulier) il est important de noter que le vecteur de *Kuhn et Tucker* λ est unique.

2.2.3 Interprétation géométrique des conditions de Karush-Kuhn et Tucker

Les conditions de Karush-Kuhn et Tucker peuvent s'interpréter géométriquement de la façon suivante (fig 2.3). Sous l'hypothèse de qualification des contraintes.

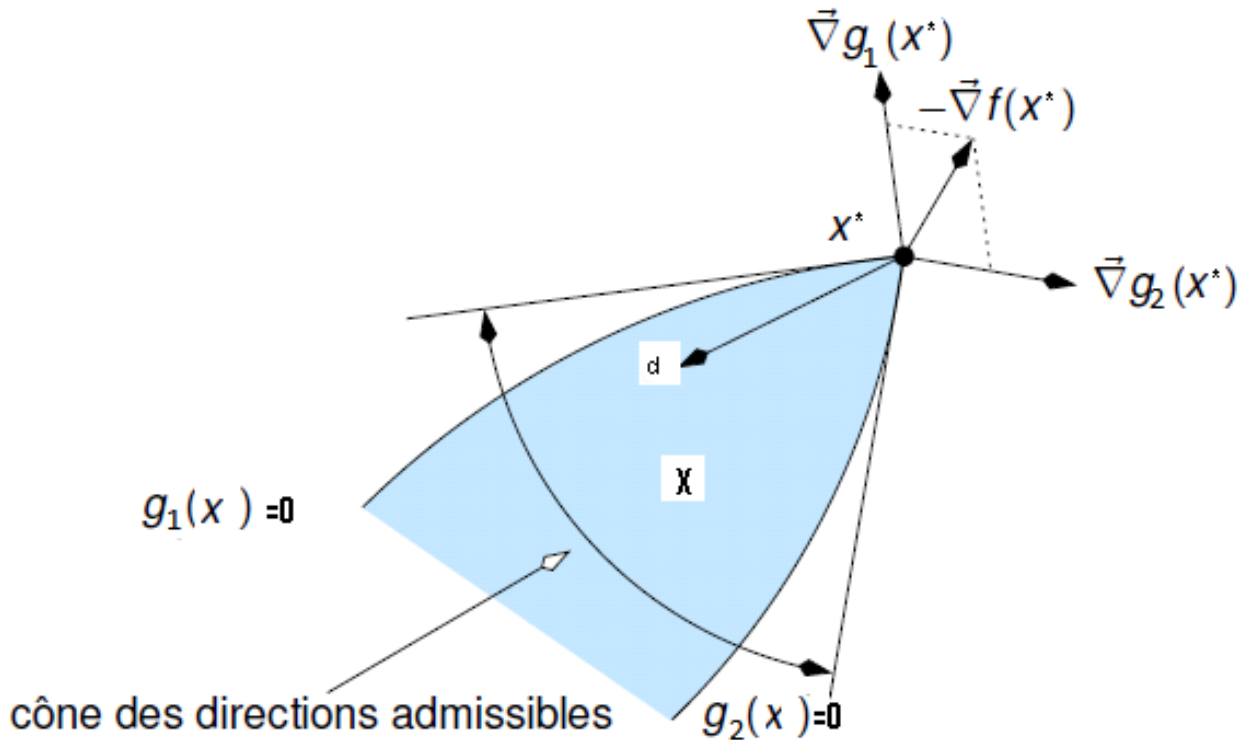


FIG. 2.3 – *Interprétation géométrique des conditions de Karush-Kuhn et Tucker*

Les conditions de Karush-Kuhn et Tucker sur un exemple à deux dimensions. Au point x^* les contraintes saturées sont les contraintes g_1 et g_2 et $\mathbb{I}_a = \{1,2\}$. L'ensemble des directions admissibles d forme un cône intersection des $|\mathbb{I}_a|$ demi-espaces d'équation :

$$\nabla g_i^T(x^*) \cdot d \leq 0 \quad (\forall i \in \mathbb{I}_a)$$

Pour que x^* soit un optimum local il faut que le vecteur $-\nabla f(x^*)$ fasse un angle obtus avec chaque direction admissible d . Il est alors équivalent de dire que le vecteur $-\nabla f(x^*)$ doit faire un angle aigu avec le gradient $\nabla g_i(x^*)$ de chaque contrainte saturée ($i \in \mathbb{I}_a$).

Dans ces conditions, $-\nabla f(x^*)$ doit effectivement s'exprimer comme combinaison linéaire à coefficients λ_i positifs des $\nabla g_i(x^*)$ pour ($i \in \mathbb{I}_a$).

Remarque 2.2.3.

1. Les conditions de Karush-Kuhn et Tucker s'étendent sans difficulté à des problèmes comportant à la fois des contraintes d'égalité et des contraintes d'inégalité de la forme

$$(P) \begin{cases} \min f(x) \\ \text{sous les contraintes} \\ g_i(x) \leq 0 & i = \{1, 2, \dots, m\} \\ h_i(x) = 0 & i = \{1, 2, \dots, p\} \\ x \in \mathbb{R}^n. \end{cases}$$

La condition nécessaire pour que x^* soit un optimum local de (P) est qu'il existe des nombres $\lambda_i \geq 0$ et μ_i non contraints en signe tels que :

$$\begin{cases} \nabla f(x^*) + \sum_{i \in \mathbb{I}} \lambda_i \nabla g_i(x^*) + \sum_{i \in \mathbb{II}} \mu_i \nabla h_i(x^*) = 0, \\ \text{et} \\ \lambda_i \nabla g_i(x^*) = 0 & (i \in \mathbb{I}) \\ h_i(x^*) = 0 & i = \{1, 2, \dots, p\} \end{cases} \quad (2.34)$$

2.2.4 Conditions suffisantes d'optimalité

Nous allons étudier maintenant des conditions suffisantes d'optimalité pour le problème(2.31).Pour cela on introduit la notion du point-col

Définition 2.2.6. "Point-col"

Soit : $\bar{x} \in X$ et $\bar{\lambda} \geq 0$.

On dit que $(\bar{x}, \bar{\lambda})$ est un point-col de $L(x, \lambda)$ si :

1. $L(\bar{x}, \bar{\lambda}) \leq L(x, \bar{\lambda}) \quad \forall x \in X$
2. $L(\bar{x}, \lambda) \leq L(\bar{x}, \bar{\lambda}) \quad \forall \lambda \geq 0$

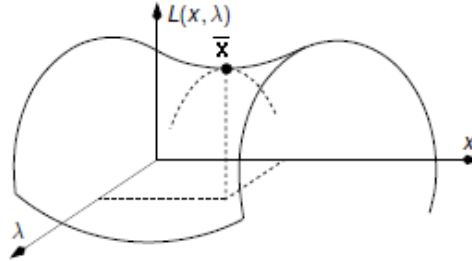


FIG. 2.4 – Illustration de la notion de point-col

Propriété 2.1. "Caractérisation d'un Point-col"

Soit $\bar{x} \in X$ et $\bar{\lambda} \geq 0$.

$(\bar{x}, \bar{\lambda})$ est un point-col pour $L(x, \lambda)$ si et seulement si :

- a) $L(\bar{x}, \bar{\lambda}) = \min_{x \in X} L(x, \lambda)$
- b) $g_i(\bar{x}) \leq 0 \quad \forall i \in \mathbb{I}$
- c) $\bar{\lambda}_i g_i(\bar{x}) = 0 \quad \forall i \in \mathbb{I}$

Théorème 2.2.3. [3]"Suffisance de la condition de point-col"

Si $(\bar{x}, \bar{\lambda})$ est un point-col de $L(x, \lambda)$ alors \bar{x} est un optimum global du problème (2.31).

Preuve.

La condition (a) du théorème précédent entraîne:

$$f(\bar{x}) + \sum_{\mathbb{I}} \bar{\lambda}_i g_i(\bar{x}) \leq f(x) + \sum_{\mathbb{I}} \bar{\lambda}_i g_i(x) \quad \forall x \in X$$

D'autre part : (c) $\implies \bar{\lambda}_i g_i(\bar{x}) = 0 \quad \forall i$

d'où : $f(\bar{x}) \leq f(x) + \sum_I \bar{\lambda}_i g_i(x) \quad \forall x \in X$ et comme $\bar{\lambda} \geq 0$ on a : $f(\bar{x}) \leq f(x), \forall x \in X$

tel que: $g_i(x) \leq 0$.

D'où \bar{x} est un optimum global du problème.

Remarque 2.2.4.

On peut utiliser aussi la matrice hessienne pour étudier la condition suffisante d'optimalité

Remarque 2.2.5.

Ce résultat est très général et s'applique à n'importe quel programme mathématique convexe, non convexe, f et g_i différentiable ou non...

Cependant, pour certains problèmes, il peut ne pas exister de point-col. C'est le cas en général, pour les problèmes non convexes.

2.2.5 Méthodes itératives

Les méthodes de pénalités

La pénalisation est un concept simple qui permet de transformer un problème d'optimisation avec contraintes en un problème ou en une suite de problèmes d'optimisation sans contrainte. C'est un concept qui a une utilité à la fois théorique et numérique. En analyse, l'approche par pénalisation est parfois utilisée pour étudier un problème d'optimisation dont les contraintes sont difficiles à prendre en compte, alors que le problème pénalisé a des propriétés mieux comprises ou plus simples à mettre en évidence. Si on a de la chance ou si la pénalisation est bien choisie, des passages à la limite parfois délicats permettent d'obtenir des propriétés du problème original (l'existence de solution par exemple). Les méthodes de pénalité constituent une famille d'algorithmes particulièrement intéressants du double point de vue de la simplicité de principe et de l'efficacité pratique. Dans cette section nous étudierons plus spécialement deux des variantes les plus utilisées : les méthodes de pénalités extérieures et les méthodes de pénalités intérieures.

Méthode de pénalité extérieure

Nous considérons le problème suivant:

$$(P1) \left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{sous les contraintes} \\ g_i(x) \leq 0 \quad (i = 1, 2, \dots, m) \\ x \in \mathbb{R}^n \end{array} \right.$$

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

Définition 2.2.7. On définit la fonction de pénalité quadratique pour le problème (P1) par:

$$p(x,c)=f(x)+c\sum_{i=1}^m \max(0,g(x))^2$$

avec c est un scalaire strictement positif, appelé le facteur de pénalisation.
Donc on aura le problème (P_c) suivant

$$(P_c) \begin{cases} \text{Minimiser} & p(x,c) = f(x) + c \sum_{i=1}^m \text{Max}\{0,g_i(x)\}^2 \\ & x \in \mathbb{R}^n \end{cases}$$

Proposition 2.2.1. *Soit la suite $\{c_k\}_k$ croissante $c_k > 0, c_k \rightarrow +\infty$*

et x^k la solution optimale du problème (P_c)

La suite $\{p(x^k), c_k\}_k$ est croissante.

Soit $\theta_k = \sum_{i=1}^m \max\{0, g_i(x^k)\}^2$ alors θ_k est décroissante.

Soit la suite $\{X^k\}_k$ solution du problème (P_c) elle converge vers la solution optimale du problème (P_1)

Preuve.

La suite $\{p(x^k), c_k\}_k$ est croissante.

$$\begin{aligned} p(x^{k+1}, c_{k+1}) &= f(x^{k+1}) + c_{k+1} \sum_{i=1}^m \max(0, g(x^{k+1}))^2 \geq f(x^{k+1}) + c_k \sum_{i=1}^m \max(0, g(x^{k+1}))^2 \\ &\geq f(x^k) + c_k \sum_{i=1}^m \max(0, g(x^k))^2 = p(x^k, c_k) \Rightarrow p(x^{k+1}, c_{k+1}) \end{aligned}$$

\Rightarrow La suite $p(x^k, c_k)_k$ est croissante. \square

θ_k est décroissante.

Soit x^k un minimum alors on aura:

$$p(x^k, c_k) = f(x^k) + c_k \theta_k \leq f(x^{k+1}) + c_k \theta_{k+1} \quad (1).$$

soit x^{k+1} le minimum donc on aura:

$$f(x^{k+1}) + c_{k+1} \theta_{k+1} \leq f(x^k) + c_k \theta_k \quad (2)$$

(1) + (2) nous donne

$$f(x^k) + c_k \theta_k + f(x^{k+1}) + c_{k+1} \theta_{k+1} \leq f(x^{k+1}) + c_k \theta_{k+1} + f(x^k) + c_k \theta_k$$

$$c_k(\theta_k - \theta_{k+1}) + c_{k+1}(\theta_{k+1} - \theta_k) \leq 0$$

$$(c_k - c_{k+1})(\theta_k - \theta_{k+1}) \leq 0 \text{ comme } (c_k - c_{k+1}) \leq 0 \text{ donc } (\theta_k - \theta_{k+1}) \geq 0 \Rightarrow \theta_k \geq \theta_{k+1}$$

$\Rightarrow \{\theta_k\}_k$ est décroissante. \square

la suite $\{X^k\}_k$ solution du problème (P_c) elle converge vers la solution optimale du problème (P_1)

Soit x^* la solution optimale du problème P_1

$$f(x^k) \leq f(x^k) + c_k \sum_{i=1}^m \max\{(0, g_i(x^k))^2\} \leq f(x^*) + c_k \sum_{i=1}^m \max\{(0, g_i(x^*))^2\} \leq f(x^*)$$

$$\Rightarrow f(x^k) \leq f(x^*)$$

comme $\{f(x^k)\}_k$ est croissante et majorée par $f(x^*)$ donc elle converge $f(x^k) \rightarrow f(\bar{x})$ quand $k \rightarrow \infty \Rightarrow x^k \rightarrow \bar{x}$

$$f(\bar{x}) \leq f(x^*) \quad (1)$$

$$\text{on a } p(x^k, c_k) - f(x^k) = c_k \theta_k \geq 0$$

$p(x^k, c_k) \leq f(x^*)$ et d'autre part on a $\{p(x^k, c_k)\}_k$ est croissante donc elle converge

$f(x^k)$ converge $\Rightarrow -f(x^k)$ converge donc $p(x^k, c_k) - f(x^k)$ converge et $c_k \theta_k$ converge

$$\theta_k = \sum_{i=1}^m \max\{(0, g_i(x^*))^2\} \rightarrow \sum_{i=1}^m \max\{(0, g_i(\bar{x}))^2\} = 0 \text{ car } g_i(\bar{x}) \leq 0 \text{ car } \bar{x} \text{ est admissible}$$

$$\Rightarrow f(x^k) \leq f(\bar{x}) \quad (2)$$

(1) + (2) $\Rightarrow f(x^k) = f(\bar{x}) \Rightarrow x^k$ converge vers la solution optimale du problème P_1 . \square

Remarque 2.2.6.

Dans le cas générale

$$(P) \left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{sous les contraintes} \\ g_i(x) \leq 0 \quad (i = 1, 2, \dots, m) \\ h_j(x) = 0 \quad (j = 1, 2, \dots, p) \\ x \in \mathbb{R}^n \end{array} \right.$$

la fonction de pénalité quadratique extérieur s'écrit sous la forme suivante:

$$p(x^k, c_k) = f(x^k) + c_k \sum_{i=1}^m \max\{0, g_i(x^k)\}^2 + c_k \sum_{j=1}^p (h_j)^2$$

Algorithme de pénalité extérieure

1. **Initialisation**

fixer $\varepsilon > 0$, choisir x^0 non admissible, choisir $\gamma > 1$ et $(c_1 \simeq 1)$

2. **Itération** $k = 1, 2, \dots$,

écrire $p(x, c_k)$ avec les contraintes non vérifiées en x^{k-1} puis résoudre

$$(Pc) \left\{ \begin{array}{l} \text{Minimiser } p(x, c) = f(x) + c \sum_{i=1}^m \max\{0, g_i(x)\}^2 \\ x \in \mathbb{R}^n \end{array} \right.$$

3. **Critère d'arrêt**

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP x^k est optimal

Si non, on pose $k = k + 1$ et on retourne à 2 avec $c_{k+1} = \gamma c_k$.

Remarque 2.2.7.

1. On commence toujours par un point non admissible.
2. Pour le choix du paramètre de pénalité on commence par des valeurs assez petites ($c_1 = 1$) ensuite l'augmente au fur et à mesure des itérations.

Méthode de pénalité intérieure

Dans certains problèmes, le fait que les itérés x^* générés par pénalisation extérieure ne soient pas admissibles peut être un inconvénient, par exemple, parce que f n'est pas définie à l'extérieur de X . On peut introduire des méthodes de pénalisation dans lesquelles les itérés x^* restent dans X . C'est ce qui a conduit *Fiacco et McCormick* (1968) à proposer d'autres méthodes de pénalisation dans lesquelles l'optimum est approché par l'intérieur (d'où le nom de méthodes de « pénalités intérieures »).

Soit le problème suivant:

$$(P1) \left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{sous les contraintes} \\ g_i(x) \leq 0 \quad (i = 1, 2, \dots, m) \\ x \in \mathbb{R}^n \end{array} \right.$$

Définition 2.2.8. On définit la fonction de pénalité intérieure pour le problème(P1) par:

$$p(x,r)=f(x)+r\sum_{i=1}^m\left(-\frac{1}{g_i(x)}\right)$$

avec r est un scalaire strictement positif, appelé le facteur de pénalisation intérieure et $g_i(x) < 0 \quad i = 1, \dots, m$.

Donc on aura le problème (P_r) suivant

$$(Pr) \left\{ \begin{array}{l} \text{Minimiser } p(x,r) = f(x) + r \sum_{i=1}^m \left(-\frac{1}{g_i(x)}\right) \\ x \in \mathbb{R}^n \end{array} \right.$$

Remarque 2.2.8.

La fonction de pénalité intérieure est appelé aussi fonction barrière elle évite de sortir du domaine ni même de franchir la frontière.

Proposition 2.2.2.

La suite $\{p(x^k, r_k)\}_k$ est une suite décroissante

Soit $\theta_k = \sum_{i=1}^m (-\frac{1}{g_i(x)})$ alors $\{\theta_k\}$ est une suite croissante

La suite $\{f(x^k)\}_k$ est une suite décroissante

Soit $\{X^k\}_k$ la solution optimale du problème(P_r) alors elle converge vers la solution optimale du problème de départ(P_1)

Algorithme de pénalité intérieure

1. **Initialisation**

fixer $\varepsilon > 0$, trouver un point x^0 admissible, choisir $\xi < 0$ et ($r_1 > 0$) on prend ($r_1 = 10$)

2. **Itération** $k = 1, 2, \dots$

$$p(x, r_k) = f(x) + r_k \sum_{i=1}^m \left(-\frac{1}{g_i(x)}\right)$$

calculer

$$\min_{x \in \mathbb{R}^n} p(x, r_k)$$

avec l'un des méthodes étudiées dans la partie précédente.

3. **Critère d'arrêt**

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP x^k est une solution optimale

Sinon, on pose $k = k + 1$ et on retourne à 2 avec $r_{k+1} = \xi r_k$.

Remarque 2.2.9.

Dans le cas générale

$$(p) \left\{ \begin{array}{l} \text{Min} f(x) \\ \text{sous les contraintes} \\ g_i(x) \leq 0 \quad i = \{1, 2, \dots, m\} \\ h_j(x) = 0 \quad j = \{1, 2, \dots, k\} \\ x \in S \end{array} \right.$$

La fonction de pénalité dans ce cas s'écrit comme suit:

$$p(x^k, c_k) = f(x^k) + \frac{1}{c} \sum_{i=1}^m \left(-\frac{1}{g_i(x)}\right) + c_k \sum_{j=1}^p (h_j)^2$$

Méthode du lagrangien augmenté

Définition 2.2.9. On définit le lagrangien augmenté pour le problème suivant:

$$(P) \begin{cases} \text{Minimiser } f(x) \\ \text{sous les contraintes} \\ h_j(x) = 0 \quad (i = 1, 2, \dots, p) \\ x \in \mathbb{R}^n \end{cases} \quad (2.35)$$

$$L_a(x, \lambda, c) = f(x) + \sum_{j=1}^p \lambda_j h_j(x) + c \sum_{j=1}^p h_j(x)^2$$

Remarque 2.2.10. Le lagrangien augmenté peut-être considéré:

- Comme le lagrangien au quel on a ajouté une pénalité quadratique
- Comme une fonction de pénalité quadratique à la quelle on a ajouté les multiplicateurs.

Remarque 2.2.11. Soit la fonction de pénalité quadratique pour le problème (P)

$$p(x,c)=f(x)+c\sum_{j=1}^p h_j(x)^2$$

$$\nabla p(x,c) = \nabla f(x) + 2c \sum_{j=1}^p h_j(x) \nabla h_j(x)$$

On a le théorème de lagrange pour le problème (P) condition nécessaire

$$\nabla f(x) + \sum_{j=1}^p \lambda_j^* \nabla h_j(x) = 0$$

$$\lambda_j^* = 2c h_j(x)$$

$$h_j(x) = \frac{1}{2c} \lambda_j^*$$

pour que x soit admissible $c \rightarrow +\infty \quad h_j(x) = 0$

Soit maintenant pour le lagrangien augmenté

$$\nabla_x L(x, \lambda, c) = \nabla f(x) + \sum_{j=1}^p \lambda_j \nabla h_j(x) + \sum_{j=1}^p 2c h_j(x) \nabla h_j(x) = \nabla f(x) + \sum_{j=1}^p (\lambda_j + 2c h_j(x)) \nabla h_j(x)$$

$$\lambda_j^* = \lambda_j + 2c h_j(x), \quad h_j(x) = \frac{\lambda_j^* - \lambda_j}{2c}$$

Si $\lambda_j^* - \lambda_j = 0$ on aura $h_j(x) = 0$

C'est-à-dire: l'admissibilité pour actualiser les multiplicateurs d'une itération à l'autre on utilise la formule suivante:

$$\lambda_j^{k+1} = \lambda_j^k + 2c_k h_j(x^k)$$

Algorithme de lagrangien augmenté

1. **Initialisation**

fixer $\varepsilon > 0$, choisir c_0 $x^{0(stable)}$ admissible, λ^0 et $(\gamma > 1)$

2. **Itération** $k = 1, 2, \dots$

calculer

$$\min_{x \in \mathbb{R}^n} L(x, \lambda^k, c_k)$$

avec l'un des méthodes étudiées dans la partie précédente.

3. **Critère d'arrêt**

Si $\|h(x^k)\| < \varepsilon$, STOP x^k est une solution optimale

actualiser le paramètre de pénalité $c_{k+1} = \gamma c_k$

actualiser le point de départ $x^{(k+1)s} = x^k$

on pose $k = k + 1$ et on retourne à 2. Sinon, actualiser les multiplicateurs $\lambda_j^{k+1} = \lambda_j^k + 2c_k h_j(x^k)$

actualiser le paramètre de pénalité $c_{k+1} = \gamma c_k$

actualiser le point de départ $x^{(k+1)s} = x^k$

on pose $k = k + 1$ et on retourne à 2.

Proposition 2.2.3. $\nabla_x L(x^*, \lambda^*, c) = \nabla_x L(x^*, \lambda^*) = 0$
avec x^* solution optimale du problème (2.35) et λ^* le vecteur des multiplicateurs correspondant

Preuve. :

On a le théorème de lagrange $\nabla_x L(x^*, \lambda^*) = 0$

$$\begin{aligned} \text{d'autre part } \nabla_x L(x^*, \lambda^*, c) &= \nabla f(x^*) + \sum_{j=1}^p \lambda_j^* \nabla h_j(x^*) + 2c \sum_{j=1}^p h_j(x^*) \nabla h_j(x^*) \\ &= \nabla f(x^*) + \sum_{j=1}^p \lambda_j^* \nabla h_j(x^*) = \nabla_x L(x^*, \lambda^*) = 0 \end{aligned}$$

Remarque 2.2.12. On a la hessienne du lagrangien augmenté

$$\begin{aligned} H_x L_a(x^*, \lambda^*, c) &= H_x L_a(x^*, \lambda^*) + 2c (Jh(x^*))^T (Jh(x^*)) + 2c \sum_{j=1}^p h_j(x^*) H_x h_j(x^*) \\ &= H_x L_a(x^*, \lambda^*) + 2c (Jh(x^*))^T (Jh(x^*)) \end{aligned}$$

Résultat d'algèbre

Soit $M_{n \times n}$ une matrice carrée symétrique et $A_{p \times n}$ une matrice de rang plein alors:
il existe c^α tel que:

$\forall c \geq c^\alpha \quad M + cA^T A$ est définie positive

$H_x L_a(x^*, \lambda^*, c) = H_x L_a(x^*, \lambda^*) + 2c (Jh(x^*))^T (Jh(x^*)) \Rightarrow H_x L_a(x^*, \lambda^*, c)$ est défini positif
pour $\forall c \geq c^\alpha \Rightarrow x^*$ est un minimum

donc pour le lagrangien augmenté on a pas besoin de faire tendre $c \rightarrow +\infty$ pour avoir la solution optimale.

on dit que c'est une fonction de pénalité exacte.

Remarque 2.2.13. Pour le problème avec contraintes inégalités on se ramène au cas avec contraintes égalités par l'ajout de variables d'écart.

Chapitre 3

Optimisation semi-infinie

Introduction

On dispose de plusieurs logiciels conviviaux pouvant résoudre des problèmes de taille considérable (finie), toute fois ces logiciels restent limités par le nombre fini de contraintes. Dans la pratique, on se trouve souvent confronté à des situations où ce nombre est infini d'où l'intérêt d'adapter les techniques de choix des méthodes classiques aux cas plus générales (où le nombre de contrainte est dénombrable (infini)). De ce concept la programmation semi-infinie est née.

3.1 Formulation d'un problème semi-infini:

Un programme semi-infini est un problème du type:

$$(P) \begin{cases} \text{Min} f(x) \\ g(x,s) \leq 0 \quad \forall s \in S \\ x \in \mathbb{R}^n \end{cases} \quad (3.1)$$

- S : est un compact de \mathbb{R}^p
- $f: \mathbb{R}^n \rightarrow \mathbb{R}$
- $g: \mathbb{R}^n \times S \rightarrow \mathbb{R}$
- Si f et g sont linéaires par rapport à x on dit qu'on a un problème semi-infini linéaire (P.S.I.L).
- Si f est convexe par rapport à x et g concave par rapport à x on dit qu'on a un problème semi-infini convexe (P.S.I.C)
- Dans les autres cas, on a un problème semi-infini non linéaire.

3.2 Exemple pratique

Un grand intérêt attire les chercheurs pour la classe des problèmes d'optimisation semi-infini, qui sont caractérisées par un nombre fini de variables et un nombre infini de contraintes de tels problèmes apparaissent par exemple dans la réduction de la pollution atmosphérique, de la solution des équations intégrales faiblement singulières, dans la distribution de probabilité, la robotique[10]...

La robotique est devenu un outil très important dans les différents domaines de recherche dans notre exemple on va s'intéresser à la planification du mouvement d'un robot humanoïde dans le cadre de valider les méthodes développées de restauration du mouvement chez un patient paraplégique comme l'indique cette figure:

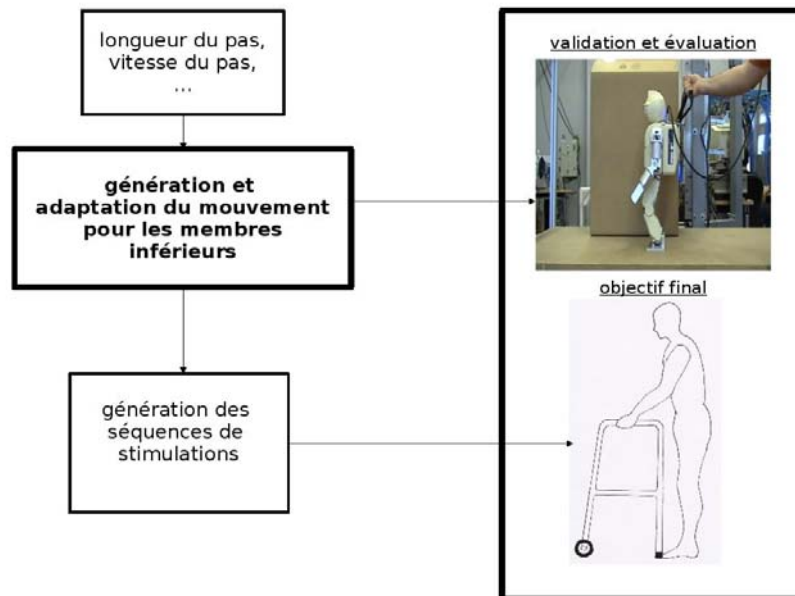


FIG. 3.1 – Les robots humanoïdes permettent de valider les méthodes développées

Définition 3.2.1. : Robot humanoïde

Un robot humanoïde est un robot dont l'apparence générale rappelle celle d'un corps humain. Généralement, les robots humanoïdes ont un torse avec une tête, deux bras et deux jambes, bien que certains modèles ne représentent qu'une partie du corps, par exemple à partir de la taille. Certains robots humanoïdes peuvent avoir un « visage », avec des « yeux » et une « bouche ».

Avant de modéliser ce problème sous forme d'un modèle mathématique on définit les

contraintes du mouvement.

nous commençons par la définition du mouvement à travers un vecteur de paramètres

$$X = [p_{0,0}, p_{0,1}; \dots; T_f]^T$$

à partir de ce vecteur de paramètres nous allons calculé les positions , vitesses et accélérations articulaires .

Définition 3.2.2. Variables articulaire :[10]

$$q_j(t) = \sum_{k=1}^N p_{j,k} \times b_k(t)$$

Les vitesses et accélérations articulaires sont obtenus par dérivations :

$$\dot{q}_j(t) = \sum_{k=1}^N p_{j,k} \times \dot{b}_k(t)$$

$$\ddot{q}_j(t) = \sum_{k=1}^N p_{j,k} \times \ddot{b}_k(t)$$

Nous considérerons qu'un mouvement optimal doit obligatoirement respecter les contraintes de butées articulaires ainsi que de vitesses articulaires limites .

$$\forall i, \forall t, q_{i,min} \leq q_i(t) \leq q_{i,max} \quad (3.2)$$

$$\forall i, \forall t, \dot{q}_{i,min} \leq \dot{q}_i(t) \leq \dot{q}_{i,max} \quad (3.3)$$

Définition 3.2.3. Equilibre ZMP :[10]

Le Zero-Moment-Point (ZMP) est le point sur la surface de contact où la résultante des moments est nulle .Si ce point reste dans la surface de sustentation le robot conserve son équilibre.

Afin d'obtenir un mouvement qui préserve l'équilibre, nous prendrons également en compte les limites du ZMP.

$$\forall t ZMP_{min} \leq ZMP(t) \leq ZMP_{max} \quad (3.4)$$

Nous considérons les contraintes des équations (3,2),(3,3),(3,4) et voulons **minimiser la durée du mouvement** on obtient le modèle d'optimisation qui consiste à trouver le vecteur des paramètres optimal X qui satisfait :

$$\left\{ \begin{array}{l} \min_{X \in \mathbb{R}} f(X) \\ g_i(X,t) \leq 0 \quad \forall i \quad \forall t \in [0,T] \\ h_j(X) = 0 \quad \forall j \end{array} \right. \quad (3.5)$$

f est la fonction objectif à minimiser ($\min \int_0^T F(X,t)dt$),

g_i est un ensemble de contraintes inégalités,

h_j est un ensemble de contraintes égalités.

Le problème posé (3,5) est un problème de programmation semi-infini car il dispose d'un nombre fini de paramètres alors que le domaine des contraintes est infini.

Chapitre 4

Méthodes de résolution d'un problème semi-infini convexe

4.1 Résolution d'un problème semi-infini convexe par la nouvelle méthode d'échange

Introduction

La méthode de discrétisation consiste à:

⇒ Trouver à partir d'un ensemble infini de points un sous-ensemble fini de points.

⇒ Résoudre le sous-problème obtenu avec des méthodes déjà connues et répéter l'opération.

A la fin on aura une suite de solutions qui va converger vers la solution optimale.

4.1.1 Problème semi-infini discrétisé

Soit (P) le problème (S.I.C) suivant:

$$(P) \left\{ \begin{array}{l} \text{Min}f(x) \\ g(x,s) \leq 0 \quad \forall s \in \Omega \\ x \in \mathbb{R}^n \end{array} \right. \quad (4.1)$$

Et soit (P_k) le sous problème suivant:

$$(P_k) \begin{cases} \text{Min}f(x) \\ g(x, s_i) \leq 0 & s_i \in E_k \\ x \in \mathbb{R}^n, E_k = \{s_1, \dots, s_k\} \subset \Omega \end{cases} \quad (4.2)$$

(P_k) est appelé problème discrétisé de P

Remarque 4.1.1. Nous nous intéressons au problème (S.I.C) qui satisfait l'ensemble des conditions suivantes:

- f est convexe et continûment différentiable dans \mathbb{R}^n
- $g(\cdot, s)$ est convexe et différentiable pour tout $s \in \Omega$ avec $\nabla_x g(x, s)$ continu dans $\mathbb{R}^n \times \Omega$
- La qualification des contraintes de Slater i.e $\exists \hat{x} \in \mathbb{R}^n$ tels que $g(\hat{x}, s) < 0$ pour tout $s \in \Omega$
- f est bornée dans l'ensemble réalisable de (P) i.e pour tout scalaire λ on a :
 $\{x \in \mathbb{R}^n | g(x, s) \leq 0, s \in \Omega; f(x) \leq \lambda\}$

4.1.2 Principe de la méthode

-) Il s'agit à chaque itération k de résoudre le problème discrétisé (P_k)
-) Supposons qu'on est à l'itération k , Pour résoudre le problème (P_k) on utilise le théorème de K.K.T suivant :
 Soit $x^* \in \mathbb{R}^n$ une solution réalisable de (P_k) , x^* est optimale ssi \exists des multiplicateurs $\lambda^* \in \mathbb{R}_+^k$ tels que (x^*, λ^*) satisfait les conditions suivantes:

$$\begin{cases} \nabla f(x) + \sum_{i=1}^k \lambda(s_i) \nabla_x g(x, s_i) = 0 \\ g(x, s_i) \lambda = 0 \\ i = 1, \dots, k \end{cases} \quad (4.3)$$

-) Soit $x^k = \{x_1^k; \dots; x_n^k\}$ la solution de (P_k)

Posons:

$$D = \{x \in \mathbb{R}^n, g(x, s) \leq 0, s \in \Omega\}$$

on a $D_k = \{x \in \mathbb{R}^n, g(x, s_i) \leq 0, s_i \in E_k\}$

$x \in D \Rightarrow x \in D_k$ on a bien $D \subset D_k$

4.1.3 Test d'optimalité

Trouver: $s_{new}^k \in \Omega$ tels que $g(x^k, s_{new}^k) > \eta$ pour $\eta > 0$

On distingue deux cas :

– **Cas 1:** Si s_{new}^k n'existe pas alors **Stop** $\Rightarrow x^k$ admissible obtenu à l'itération k est optimale

– **Cas 2:** On doit chercher une autre solution réalisable optimale et pour cela

posons $\bar{E}_{k+1} := E_k \cup s_{new}^k$

En considérant le nouveau problème discrétisé $P(\bar{E}_{k+1})$ suivant:

$$P(\bar{E}_{k+1}) \begin{cases} \text{Min}f(x) \\ g(x, s_i) \leq 0 \quad s_i \in \bar{E}_{k+1} \\ x \in \mathbb{R}^n \end{cases} \quad (4.4)$$

Soit l'ensemble

on a $D_{k+1} = \{x \in \mathbb{R}^n, g(x, s_i) \leq 0, s_i \in \bar{E}_{k+1}\}$

$x \in D \Rightarrow x \in D_{k+1}$ on a bien $D \subset D_{k+1}$

Soit $x^{k+1} = \{x_1^{k+1}; \dots; x_n^{k+1}\}$ la solution du problème $P(\bar{E}_{k+1})$

On définit $E_{k+1} := \{s \in \bar{E}_{k+1} | s \in \Omega \text{ ou } \lambda_{k+1}(s) > 0\}$

Passons à l'itération suivante i.e $k:=k+1$

On utilise le même test que précédemment pour vérifier l'optimalité de x^{k+1} pour (4.4) et ainsi de suite jusqu'à l'obtention de la solution x^* du problème (4.1)

Remarque 4.1.2 (1). A l'itération k de l'Algorithme nous résolvons le sous problème $P(E_k)$ et on définit à nouveau l'ensemble $E_{k+1}:=\{s \in E_k \cup \{s_{new}^k | s \in \Omega \text{ ou } \lambda_{k+1}(s) > 0\}\}$ avec s_{new}^k qui satisfait le critère: $g(x^k, s_{new}^k) > \eta$, par conséquent cet algorithme ne fait pas appel à la recherche du ε -maximum global donné par la formule suivante: $s^k \in \varepsilon\text{-argmax}\{g(x^k, s) | s \in \Omega\}$ ce qui est exigé dans une grande majorité des méthodes de résolution. cette méthode garde seulement les contraintes actives avec les multiplicateurs positifs associés

Théorème 4.1.1 (1). Soit $(x^m)_m$ la suite des solutions des problèmes discrétisés: $(P_1), (P_2), \dots, (P_m), (P_{m+1})$. avec: $S_1 \subset S_2 \subset \dots \subset S_m \subset S_{m+1} \subset \dots \subset S$ du problème primal. la suite $(x^m)_m$ converge vers la solution optimale x^* du problème primal.

4.1.4 Algorithme de résolution de (S.I.C) par la nouvelle méthode d'échange

1. Choisir un sous ensemble fini E_k de Ω et un nombre positif $\eta > 0$ (trés petit).
2. Résoudre le problème discrétisé P_k ou

$$(P_k) \begin{cases} \text{Min}f(x) \\ g(x, s_i) \leq 0 \\ x \in \mathbb{R}^n, E_k = \{s_1, \dots, s_k\} \subset \Omega \end{cases} \quad s_i \in E_k \quad (4.5)$$

Soit x^k sa solution.

3. Trouver $s_{new}^k \in \Omega$ tels que $g(x^k, s_{new}^k) > \eta$ $\eta > 0$
4. Si s_{new}^k n'existe pas aller vers 5 sinon aller vers 6
5. écrire: x^k est la solution optimale de P, aller vers 9

6. Posons $\bar{E}_{k+1} = E_k \cup s_{new}^k$, et résoudre le problème suivant:

$$P(\bar{E}_{k+1}) \begin{cases} \text{Min}f(x) \\ g(x, s_i) \leq 0 \quad s_i \in \bar{E}_{k+1} \\ x \in \mathbb{R}^n \end{cases} \quad (4.6)$$

7. Définir $E_{k+1} := \{s \in \bar{E}_{k+1} | s \in \Omega \text{ ou } \lambda_{k+1}(s) > 0\}$

8. Posons $k = k + 1$ et aller vers 3

9. Fin

4.1.5 Exemples numériques

Exemple 1

Soit le problème semi-infini suivant:

$$(P) \begin{cases} \text{Min}(1 - x) \\ x^2 - 1 - s^2 + s \leq 0 \quad s \in [0, 1] \\ x \geq 0 \end{cases} \quad (4.7)$$

Posons: $E_0 = \{1\}, \eta = 10^{-8}$

Donc le problème discrétisé correspondant est:

$$(P_1) \begin{cases} \text{Min}(1 - x) \\ x^2 - 1 \leq 0 \\ x \geq 0 \end{cases} \quad (4.8)$$

Nous résolvons le problème (P_1) en appliquant le théorème de (K-K-T)

$$\begin{cases} -1 + \lambda(2x) = 0 & (1) \\ \lambda(x^2 - 1) = 0 & (2) \end{cases} \quad (4.9)$$

1^{er} cas :

$\lambda = 0$ on remplace dans (1) $\Rightarrow -1=0$ (impossible)

2^{ème} cas :

$x^2 - 1 = 0 \Rightarrow x = 1$ et $x = -1$

pour $x=1 \Rightarrow \lambda = \frac{1}{2}$

pour $x=-1 \Rightarrow \lambda = -\frac{1}{2}$ (impossible) les multiplicateurs doivent être positifs

donc on sort avec la solution $x=1$ avec $\lambda = \frac{1}{2}$

Calcul de s_{new}

Soit la discrétisation uniforme suivante $\{0, \frac{1}{2}, 1\}$

- On a : $g(1,0) = 0, g(1, \frac{1}{2}) = \frac{1}{4} > \eta$
donc $s_{new} = \frac{1}{2}$.

$$\begin{cases} \min(1-x) \\ (x^2 - 1) \leq 0 \\ (x^2 - 1 + \frac{1}{4}) \leq 0 \end{cases} \quad (4.10)$$

On trouve la solution

$$x^1 = \frac{\sqrt{3}}{2} \quad \lambda^1 = \frac{\sqrt{3}}{3}$$

-On a $g(\frac{\sqrt{3}}{2}, 0) = -1 \leq 0$

On passe à la discrétisation uniforme suivante $\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$
on vérifie les contraintes pour x^1 avec cette discrétisation, elles sont toutes vérifiées. On passe à une autre discrétisation on remarque que à n'importe quel nombre de points de la discrétisation on ne trouve pas s_{new} donc la solution est $x^1 = \frac{\sqrt{3}}{2}$ est optimale i.e on peut

vérifier facilement que $\max_{s \in [0,1]} g(x^1, s) \leq 0$

Donc c'est la solution optimale

Exemple 2

Soit le problème semi-infini suivant:

$$(P) \begin{cases} \text{Min}(2x + 1) \\ x + s - 1 \geq 0 \quad s \in [0,1] \\ x \geq 0 \end{cases} \quad (4.11)$$

\Downarrow

$$(P) \begin{cases} \text{Min}(2x + 1) \\ -x - s + 1 \leq 0 \quad s \in [0,1] \\ x \geq 0 \end{cases} \quad (4.12)$$

Posons: $E_0 = \{1\}, \eta = 10^{-8}$

Donc le problème discrétisé correspondant est:

$$(P_1) \begin{cases} \text{Min}(2x + 1) \\ -x \leq 0 \end{cases} \quad (4.13)$$

qui a pour solution: $x=0$

Calcul de s_{new}

Soit la discrétisation uniforme suivante $\{0, \frac{1}{2}, 1\}$

-On $g(0,0) = 1 > \eta$, alors $s_{new}^k = 0$

Posons $\overline{E}_1 = E_0 \cup \{0\} = [0,1]$, et considérons le problème discrétisé:

$$(P_2) \begin{cases} \text{Min}(2x + 1) \\ -x \leq -1 \\ -x \leq 0 \end{cases} \quad (4.14)$$

qui a pour solution $x=1$

$$\text{-On } ag(1, \frac{1}{2}) = -\frac{1}{2} \leq 0$$

$$g(1, \frac{1}{4}) = -\frac{1}{4} \leq 0$$

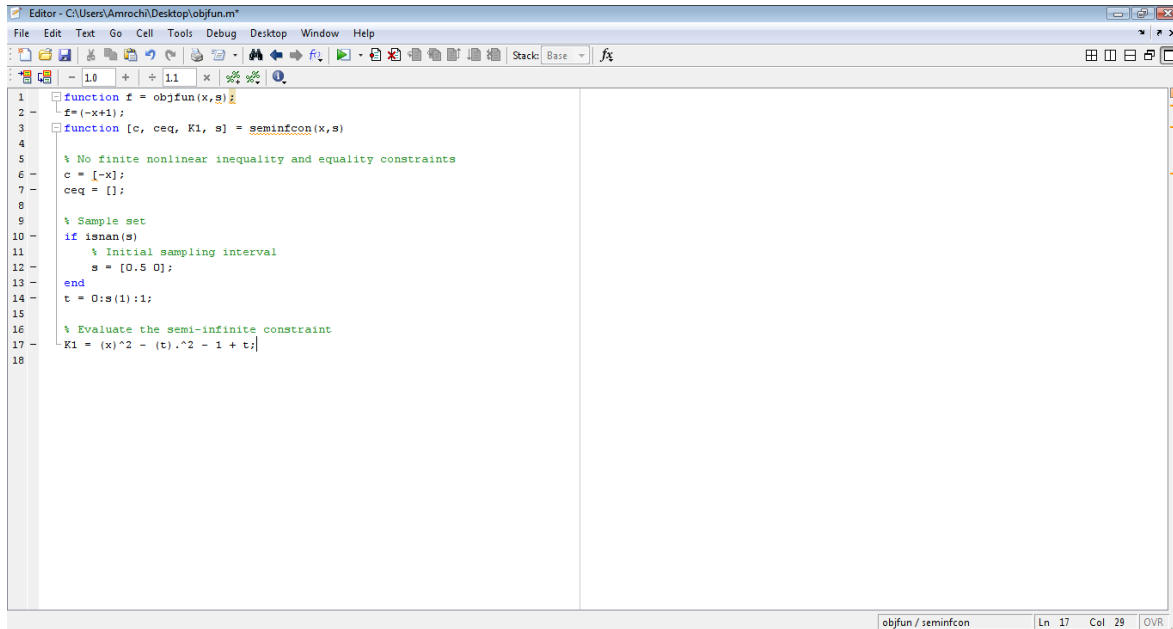
On remarque que à n'importe quel nombre de points de la discrétisation on ne trouve pas de $s_{new} \Rightarrow$ donc $x^* = 1$ est la solution optimale optimale du problème (P)

4.1.6 programmes sur MATLAB

On vérifie les résultats obtenus dans (1) et (2) sur Matlab en utilisant la fonction **fseminf**(prédéfinie dans matlab)

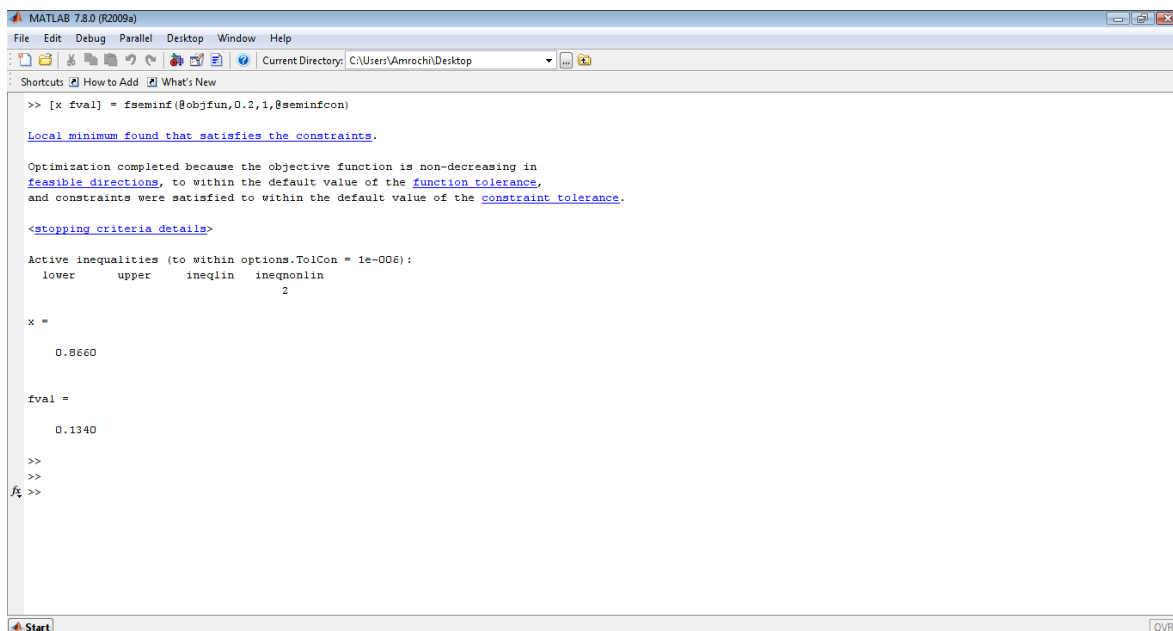
Exemple 1

programme



```
1 function f = objfun(x,s)
2     f = (-x+1);
3 function [c, ceq, K1, s] = seminfcon(x,s)
4
5     % No finite nonlinear inequality and equality constraints
6     c = [-x];
7     ceq = [];
8
9     % Sample set
10    if isempty(s)
11        % Initial sampling interval
12        s = [0.5 0];
13    end
14    t = 0:s(1):1;
15
16    % Evaluate the semi-infinite constraint
17    K1 = (x)^2 - (t).^2 - 1 + t;
18
```

Exécution



```
MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\Amroch\Desktop
Shortcuts: How to Add What's New

>> [x fval] = fseminf(@objfun,0.2,1,@seminfcon)

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

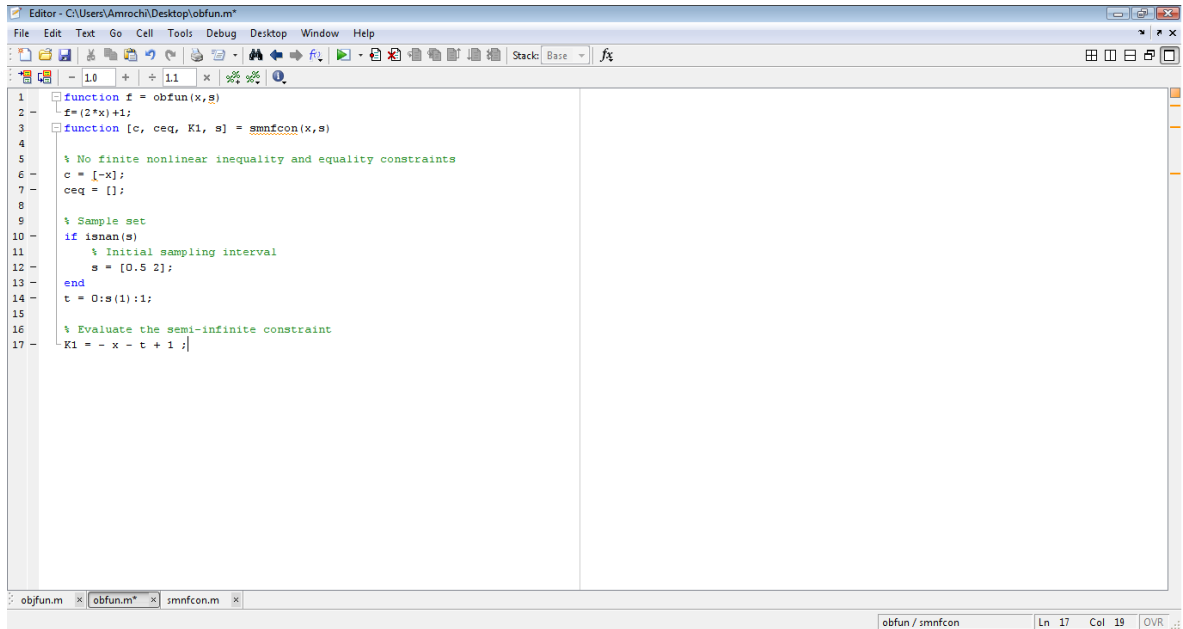
Active inequalities (to within options.TolCon = 1e-006):
    lower    upper    ineqlin    ineqnonlin
         2
x =
    0.8660

fval =
    0.1340

>>
fx >>
```

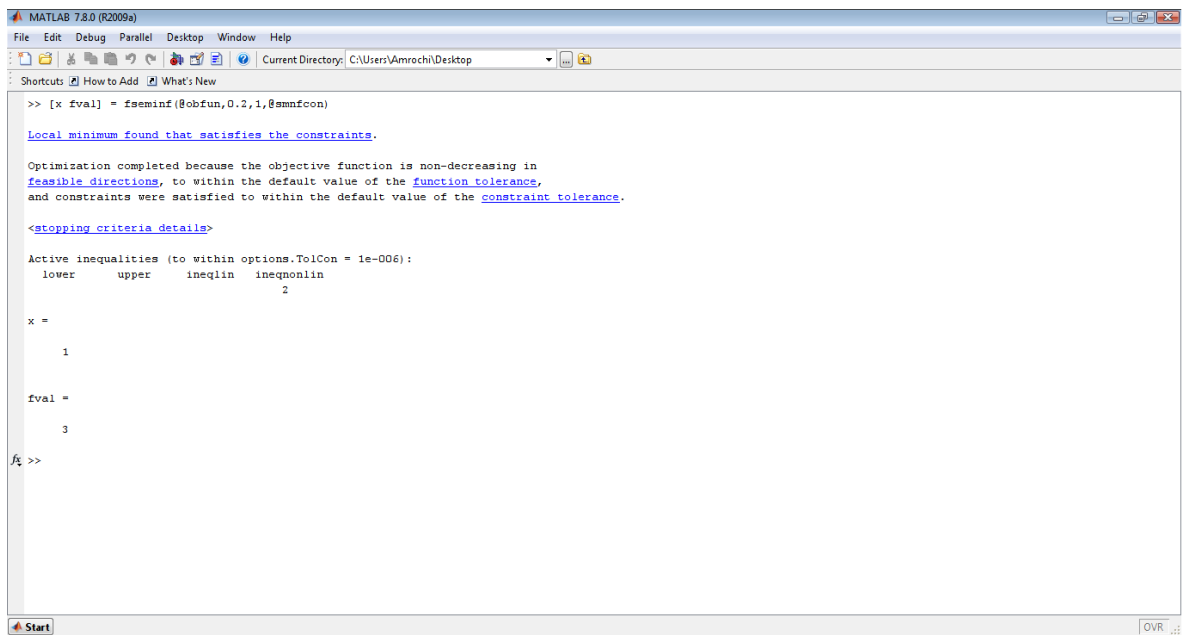
Exemple 2

programme



```
1 function f = obfun(x,s)
2     f = (2*x)+1;
3     function [c, ceq, K1, s] = smnfcon(x,s)
4
5         % No finite nonlinear inequality and equality constraints
6         c = [-x];
7         ceq = [];
8
9         % Sample set
10        if isnan(s)
11            % Initial sampling interval
12            s = [0.5 2];
13        end
14        t = 0:s(1):1;
15
16        % Evaluate the semi-infinite constraint
17        K1 = - x - t + 1 ;
```

Exécution



```
>> [x fval] = fseminf(@obfun,0.2,1,@smnfcon)

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

Active inequalities (to within options.TolCon = 1e-006):
    lower    upper    ineqlin    ineqnonlin
         2

x =

     1

fval =

     3

f>>
```

4.2 Résolution d'un programme semi-infini convexe par la méthode des coupes

position du problème

Soit le programme semi-infini suivant:

$$(P_c) \begin{cases} \text{Min} f(X) \\ g_i(X,s) \geq b_i(s) \quad i = 1, \dots, p \quad s \in S \subset \mathbb{R}^k \\ X \in \mathbb{R}^n \end{cases}$$

f est une fonction convexe et $g_i(X,s)$, $i=1, \dots, p$ sont des fonctions concaves par rapport à la variable X

supposons que les fonctions f, g_i et b_i sont de classe \mathcal{C}^2 .

Le programme peut s'écrire sous d'autre forme:

$$(\dot{P}_c) \begin{cases} \text{Min} \lambda \\ \lambda - f(X) \geq 0 \\ g_i(X,s) \geq b_i(s) \quad i = 1, \dots, p \quad s \in S \subset \mathbb{R}^k \\ X \in \mathbb{R}^n \quad \lambda \in \mathbb{R} \end{cases}$$

D'une façon générale, on peut résoudre le problème suivant:

$$(\ddot{P}_c) \begin{cases} \text{Min} CY \\ g_i(Y,s) \geq b_i(s) \quad i = 0, \dots, p \quad s \in S \subset \mathbb{R}^k \\ X \in \mathbb{R}^{n+1} \end{cases}$$

où:

$$\begin{aligned} C &= (1, 0, \dots, 0) \in \mathbb{R}^{n+1} \\ Y &= (\lambda, x_1, \dots, x_n) \in \mathbb{R}^{n+1} \\ g_0(Y, s) &= \lambda - f(X) \\ b_0(s) &= 0 \\ g_i(Y, s) &\geq b_i(s) \quad i = 1, \dots, p \end{aligned}$$

Les trois écritures sont équivalentes.

Dans la suite on utilisera la forme (\check{P}_c)

posons:

$$D = \{Y \in \mathbb{R}^{n+1} / g_i(Y, s) \geq b_i(s) \quad i = 0, \dots, p, s \in S \subset \mathbb{R}^k, |S| = \infty\},$$

qui est l'ensemble des solutions réalisables du programme (\check{P}_c)

Le principe de la méthode

La méthode des coupes appliquée pour la résolution d'un programme semi-infini convexe consiste à résoudre le programme dont le domaine des solutions réalisable D sur un ensemble H plus grand tel que $D \subset H$ ou la résolution est plus facile puis faire des coupes linéaires sur H qui ne coupe jamais le domaine D jusqu'à l'obtention de la solution optimale. Comme les contraintes ne sont pas toujours linéaires donc nous allons linéariser les sous programmes.

Pour commencer choisissons donc une matrice $A_{(p+1) \times (n+1)}$ et un vecteur b de $(n+1)$ colonnes tel que :

$$H = \{Y \in \mathbb{R}^{n+1} / AY \geq b\} \supset D,$$

4.2.1 La résolution des programmes $(\check{P}_c)^k$

Résoudre d'abord le programme initial suivant:

$$(\check{P}_c)^0 \begin{cases} \text{Min} CY \\ Y \in H \end{cases}$$

Soit Y^0 sa solution optimale.

calculons:

$$L_c^0 = \min_{i=0,\dots,p} [\min_{s \in S} (g_i(Y^0, s) - b_i(s))] \\ = \bar{g}_{i_0}(Y^0, s^0) - b_{i_0}(s^0).$$

Soit Y^k la solution de $k^{\text{ème}}$ programme $(\ddot{P}_c)^k$, et notons D^k son ensemble réalisable.

calculons:

$$L_c^k = \min_{i=0,\dots,p} [\min_{s \in S} (g_i(Y^k, s) - b_i(s))] \\ = g_{i_k}(Y^k, s^k) - b_{i_k}(s^k).$$

4.2.2 Teste d'optimalité de la solution Y^k du programme (\ddot{P}_c) pour le programme (P_c)

Après avoir calculé:

$$L_c^k = g_{i_k}(Y^k, s^k) - b_{i_k}(s^k).$$

on peut distinguer deux cas:

Cas 1: Soit $L_c^k \geq 0$, Alors on a le théorème suivant:

Théorème 4.2.1. *:[4] Si la solution optimale Y^k du programme $(\ddot{P}_c)^k$ est telle que:*

$$L_c^0 = \min_{i=0,\dots,p} [\min_{s \in S} (g_i(Y^0, s) - b_i(s))] \geq 0.$$

Alors Y^k est une solution optimale du $(\ddot{P}_c)^k$

Preuve. :

1. **Réalisable:**

$$\text{Si: } L_c^k \geq 0 \Rightarrow \min_{i=0,\dots,p} [\min_{s \in S} (g_i(Y^k, s) - b_i(s))] \geq 0.$$

$$\Rightarrow \min_{s \in S} (g_i(Y^k, s) - b_i(s)) \geq 0, \forall i = 0, \dots, p.$$

$$\Rightarrow (g_i(Y^k, s) - b_i(s)) \geq 0, \forall i = 0, \dots, p, \forall s \in S \subset \mathbb{R}^k.$$

donc:

$$Y^k \notin D \tag{4.15}$$

2. **Optimale:**

Soit $V(\check{P}_c)$ et $V(\check{P}_c^k)$ les valeurs optimales des programmes (\check{P}_c) et (\check{P}_c^k) respectivement.

Comme Y^k est réalisable pour (\check{P}_c) , alors:

$$V(\check{P}_c) \leq CY^k = V[(\check{P}_c^k)] \tag{4.16}$$

D'autre part on a: $D \subset D^k$, donc:

$$CY^k = V[(\check{P}_c^k)] \leq V(\check{P}_c) \tag{4.17}$$

de (4.14), (4.15), (4.16) on peut conclure que effectivement Y^k est une solution optimale de \check{P}_c ■

Cas 2: Si $L_c^k < 0$:

dans ce cas la solution Y^k n'est pas réalisable pour (\check{P}_c) , c'est-à-dire: $Y^k \notin D$
donc il faut ajouter la contrainte violée par Y^k , comme $g_{i_k}(Y^k, s^k) - b_{i_k}(s) < 0$ est convexe, nous allons linéariser à chaque fois la contrainte à ajouter, et ça pour avoir à faire à chaque itération k, à un programme linéaire.

4.2.3 Construction de la coupe

Au lieu d'ajouter la contrainte convexe:

$$g_{i_k}(Y^k, s^k) - b_{i_k}(s) \geq 0 \quad (4.18)$$

qui est le demi-espace délimité par la courbe convexe: $g_{i_k}(Y, s^k) - b_{i_k}(s)$ au programme $(\check{P}_c)^k$, on ajoute la contrainte:

$$g_{i_k}(Y^k, s^k) + [\nabla_y g_{i_k}(Y^k, s^k)]^T (Y - Y^k) \geq b_{i_k}(s) \quad (4.19)$$

qui est linéaire géométriquement, elle représente le demi-espace délimité par la tangente de la courbe précédente au point Y^k .

Pour le domaine réalisable D^k du programme $(\check{P}_c)^k$, cette contrainte représente une coupe linéaire qui élimine la solution Y^k .

Posons donc $(\check{P}_c)^{k+1}$ le programme résultant, et notons D^{k+1} son ensemble réalisable.

Nous allons montrer que:

$$D \subset \dots \subset D^{k+1} \subset D^k \subset \dots \subset H.$$

Où D^k est l'ensemble des $Y \in \mathbb{R}^{n+1}$ qui vérifient les contraintes linéaires du programme $(\check{P}_c)^k$.

4.2.4 Les coupes ne coupent jamais le domaine D

Il s'agit de montrer que le domaine D est un sous ensemble du domaine $D^{k+1}, \forall k$.

Pour le programme initiale (P_c^0) on a: $D \subset H$ par construction.

supposons donc que: $D \subset D^k$, on a aussi:

$$D^{k+1} = D^k \cap \{Y \in \mathbb{R}^{n+1} / g_{i_k}(Y^k, s^k) + [\nabla_y g_{i_k}(Y^k, s^k)]^T (Y - Y^k) \geq b_{i_k}(s^k)\}.$$

Soit $Y \in D$, donc: $Y \in D^k$ par hypothèse.

D'autre part, comme la courbe: $g_{i_k}(Y, s^k)$ est convexe, alors elle est au dessous de n'importe

quelle tangente en un point Y^k , c'est à dire:

$$g_{i_k}(Y, s^k) \leq g_{i_k}(Y^k, s^k) - b_{i_k}(s^k) + [\nabla_y g_{i_k}(Y^k, s^k)]^T (Y - Y^k), \forall Y^k \in \mathbb{R}^{k+1} \quad (4.20)$$

mais on a pris: $Y \in D$, donc: $g_{i_k}(Y, s^k) \geq 0$, ça implique, d'après l'inégalité (4.19) que:

$$\begin{aligned} & g_{i_k}(Y^k, s^k) - b_{i_k}(s^k) + [\nabla_y g_{i_k}(Y^k, s^k)]^T (Y - Y^k) \geq 0 \\ \Rightarrow Y \in D^k \cap \{Y \in \mathbb{R}^{n+1} / g_{i_k}(Y^k, s^k) - b_{i_k}(s^k) + [\nabla_y g_{i_k}(Y^k, s^k)]^T (Y - Y^k) \geq 0\} &= D^{k+1} \\ \Rightarrow Y \in D^{k+1}. D \subset D^{k+1} \end{aligned}$$

Donc les coupes ne coupent jamais le domaine D , seulement on réduit au fur et à mesure le domaine H et ainsi le domaine D^k , $\forall k$, en s'approchant du domaine D et de la solution du problème (\check{P}_c) .

Après avoir défini le problème $(\check{P}_c)^{k+1}$, soit Y^{k+1} sa solution, posons: $k = k + 1$, et revenant au teste d'optimalité, et ainsi de suite jusqu'à l'obtention de la solution optimale du problème (\check{P}_c)

4.2.5 La convergence

La convergence de la méthode des coupes donnée ci-dessus pour le programme semi-infini convexe (P''_c) est assurée par le théorème suivant:

Théorème 4.2.2 (4). *Tout point d'accumulation de la suite $(Y^k)_k$ engendrée par*

la procédure précédente, est une solution optimale du programme (P''_c) .

Preuve. L'existence d'au moins un point d'accumulation est donnée par le théorème de Bolzano-Weierstrass Soit Y^* un point d'accumulation de cette suite, nous allons montrer que Y^* est réalisable optimale pour (P''_c) réalisable:

Y^* un point d'accumulation de $(Y^k)_k$ dans \mathbb{R}^{n+1} qui est complet, donc il existe une sous-suite $(Y^{k_j})_j$ de $(Y^k)_k$ qui converge vers Y^* . Supposons que Y^* n'est pas réalisable pour (P''_c) alors: $\exists s^* \in S$ et $i_* \in \{0, \dots, p\}$ tels que:

$$g_{i_*}(Y^*, s^*) < b_{i_*}(s^*). \quad (4.21)$$

Comme g_{i^*} est continue (par hypothèse), alors : $\exists N \in \mathbb{N}$ tel que :

$$g_{i^*}(Y^{kj}, s^*) < b_{i^*}(s^*), \forall k_j > N. \quad (4.22)$$

sinon, en passant à la limite on aura une contradiction avec l'inégalité (4.20) on a de plus :

$L_C^{kj} = g_{i_{k_j}}(Y^{kj}, s^{kj}) - b_{i_{k_j}}(s^{kj}) \leq g_{i^*}(Y^{kj}, s^*) < b_{i^*}(s^*) < 0$. Donc $\exists \alpha > 0$ tel que:

$$b_{i_{k_j}}(s^{kj}) - g_{i_{k_j}}(Y^{kj}, s^{kj}) \geq \alpha > 0. \quad (4.23)$$

et ça en utilisant la propriété d'Archimède sur \mathbb{R} .

D'autre part on a, du fait que $(Y^{k_j})_j$ est une solution optimale du programme $(\check{P}_c)^{k_j+1}$:

$$g_{i_{k_j}}(Y^{k_j}, s^{k_j}) - b_{i_{k_j}}(s^{k_j}) + \nabla_y g_{i_{k_j}}(Y^{k_j}, s^{k_j})(Y^{k_j+1} - Y^{k_j}) \geq 0. \quad (4.24)$$

En faisant la somme (4.22) + (4.23) on aura:

$$\nabla_y g_{i_{k_j}}(Y^{k_j}, s^{k_j})(Y^{k_j+1} - Y^{k_j}) \geq \alpha > 0. \quad (4.25)$$

Si on pose :

$$K = \max_{Y \in H} \max_{s \in S} \|\nabla_y g_{i_{k_j}}(Y, s)\|.$$

On aura:

$$0 < \alpha \leq \nabla_y g_{i_{k_j}}(Y^{k_j}, s^{k_j})(Y^{k_j+1} - Y^{k_j})$$

$$\leq \|\nabla_y g_{i_{k_j}}(Y^{k_j}, s^{k_j})\| \|Y^{k_j+1} - Y^{k_j}\|$$

$$\leq K \|Y^{k_j+1} - Y^{k_j}\|, \forall k_j > N.$$

$$\Rightarrow \|Y^{k_j+1} - Y^{k_j}\| \geq \frac{\alpha}{K} > 0, \forall k_j > N$$

Mais ça contredit le fait que $(Y^{k_j})_j$, est une suite de Cauchy dans \mathbb{R}^{n+1} , donc Y^* est réalisable pour (\check{P}_c) , c'est à dire:

$$Y^* \in D. \quad (4.26)$$

2. Optimale:

D'après (4.25) on a:

$$V(\check{p}_c) \leq CY^*. \quad (4.27)$$

On a aussi le domaine réalisable D du programme (\check{p}_c) est un sous ensemble de l'ensemble D^{k_j} , qui est le domaine des solutions réalisables du programme (\check{P}^{k_j}) qui correspond à la sous-suite $(Y^j)_j$ donc :

$$CY^{k_j} \leq V(\check{P}_c), \forall K_j > N.$$

en passant à la limite on aura:

$$CY^* \leq V(\check{P}_c). \quad (4.28)$$

(4.25), (4.26), (4.27) $\Rightarrow Y^*$ est une solution optimale du programme (\check{P}_c) . ■

4.2.6 Exemple numérique

Soit le programme convexe semi-infini suivant:

$$(P_c) \left\{ \begin{array}{l} \min(x_1)^2 - 4x_2. \\ -x_1 \cos s - x_2 \sin s + 1 \geq 0 \quad s \in [0, \pi] \\ (x_1, x_2) \in \mathbb{R}^2 \end{array} \right.$$

qui peut s'écrire sous les deux formes suivantes :

$$(\check{P}_c) \left\{ \begin{array}{l} \min \lambda. \\ \lambda - (x_1)^2 + 4x_2 \geq 0 \\ -x_1 \cos s - x_2 \sin s + 1 \geq 0 \quad s \in [0, \pi] \\ (x_1, x_2, \lambda) \in \mathbb{R}^3 \end{array} \right.$$

$$(\ddot{P}_c) \left\{ \begin{array}{l} \min y_1. \\ y_1 - (y_2)^2 + 4y_3 \geq 0 \\ -y_2 \cos s - y_3 \sin s + 1 \geq 0 \quad s \in [0, \pi] \\ (y_1, y_2, y_3) \in \mathbb{R}^3 \end{array} \right.$$

Nous allons donc utiliser la méthode des coupes décrite ci-dessus pour résoudre (\ddot{P}_c) .

choisissons l'ensemble:

$$H = \{(y_1, y_2, y_3) \in \mathbb{R}^3 / y_1 - (y_2)^2 + 4y_3 \geq 0, -1 \leq y_2 \leq 1, y_3 \leq 2\}$$

On a bien :

$$D \subset H = \{(y_1, y_2, y_3) \in \mathbb{R}^n / y_1 - (y_2)^2 + 4y_3 \geq 0, -y_2 \cos s - y_3 \sin s + 1 \geq 0, s \in [0, \pi]\}.$$

Soit le programme correspondant à H suivant:

$$(\ddot{P}_c)^0 \left\{ \begin{array}{l} \min y_1. \\ y_1 - (y_2)^2 + 4y_3 \geq 0 \\ y_2 \geq -1 \\ -y_2 \geq -1 \\ -y_3 \geq -2 \\ (y_1, y_2, y_3) \in \mathbb{R}^3 \end{array} \right.$$

La solution de $(\ddot{P}_c)^0$ est: $(y_1^0, y_2^0, y_3^0) = (-8, 0, 2)$.

calculons:

$$L_c^0 = \min_{s \in [0, \pi]} [y_1^0 - (y_2^0)^2 + 4y_3^0, -y_2^0 \cos s - y_3^0 \sin s + 1]$$

$$= \min_{s \in [0, \pi]} [0, -2 \sin s + 1] = -1 < 0 \text{ pour } s_0 = \frac{\pi}{2}$$

donc $(-8, 0, 2)$ n'est pas la solution optimale pour (\ddot{P}_c) .

donc la non optimalité de cette solution entraîne l'ajout de la contrainte linéaire

(coupe linéaire) au programme $(\ddot{P}_c)^0$ qui est :

$$[-y_2^0 \cos s - y_3^0 \sin s + 1] + (0, -\cos s, -\sin s) \begin{pmatrix} y_1 + 8 \\ y_2 - 8 \\ y_3 - 2 \end{pmatrix} \geq 0.$$

$$\Leftrightarrow -1 + (-1)(y_3 - 2) \geq 0.$$

$$\Leftrightarrow y_3 \leq 1.$$

Soit le programme résultant $(\ddot{P}_c)^1$:

$$(\ddot{P}_c)^1 \left\{ \begin{array}{l} \min y_1. \\ y_1 - (y_2)^2 + 4y_3 \geq 0 \\ y_2 \geq -1 \\ -y_2 \geq -1 \\ -y_3 \geq -2 \\ -y_3 \geq -1 \\ (y_1, y_2, y_3) \in \mathbb{R}^3 \end{array} \right.$$

sa solution est: $(y_1^1, y_2^1, y_3^1) = (-4, 0, 1)$

Calculons:

$$L_c^1 = \min_{s \in [0, \pi]} [y_1^1 - (y_2^1)^2 + 4y_3^1, -y_2^1 \cos s - y_3^1 \sin s + 1]$$

$$= \min_{s \in [0, \pi]} [0, -\sin s + 1] = 0 \geq 0.$$

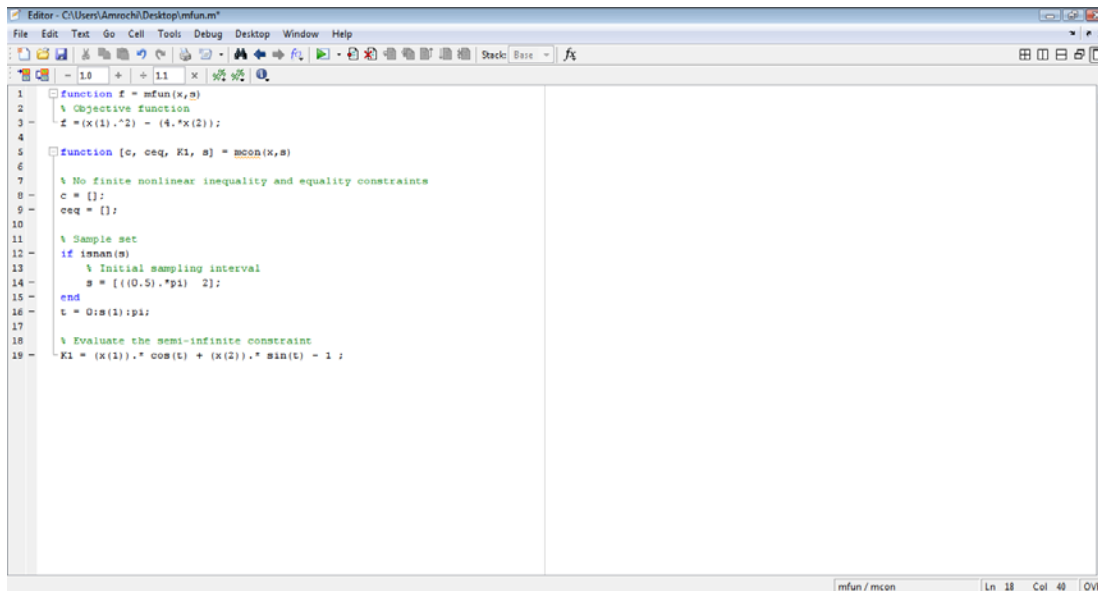
comme : $L_c^1 \geq 0$, alors $(-4, 0, 1)$ est la solution optimale du programme (\check{P}_c) , et donc de (P_c) .

4.2.7 programmes sur MATLAB

verification des resultat avec matlab

Exemple 1

programme

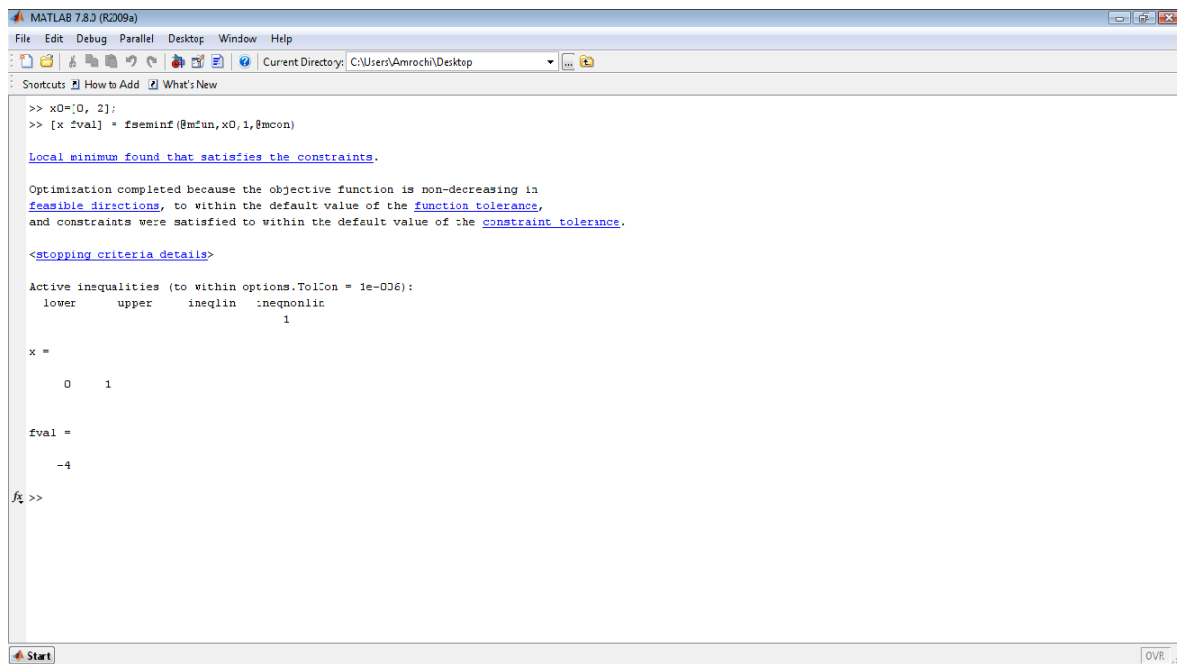


```

1 function f = mfun(x,s)
2   % Objective function
3   f = (x(1).^2) - (4.*x(2));
4
5 function [c, ceq, K1, s] = mcon(x,s)
6
7   % No finite nonlinear inequality and equality constraints
8   c = [];
9   ceq = [];
10
11  % Sample set
12  if isempty(s)
13     % Initial sampling interval
14     s = [(0.5).*pi] 2);
15  end
16  t = 0:s(1):pi;
17
18  % Evaluate the semi-infinite constraint
19  K1 = (x(1)).* cos(t) + (x(2)).* sin(t) - 1 ;

```

Exécution



```
MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\Amrochi\Desktop
Shortcuts: How to Add What's New

>> x0=[0, 2];
>> [x fval] = fseminf(@mfun,x0,1,@mcon)

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible direction, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

Active inequalities (to within options.TolCon = 1e-036):
   lower    upper   ineqlin  ineqnonlin
       1
      1

x =
    0    1

fval =
   -4

fx >>
```

Conclusion

Nous nous sommes intéressés dans notre travail à une classe particulière de problèmes d'optimisation, à savoir: les problèmes mathématiques semi-infinis.

Nous avons montré comment un problème de décision qui est **La planification de trajectoire d'un robot humanoïde** peut se ramener à un programme mathématique semi-infini.

Nous avons aussi présenté deux méthodes de résolution, en tenant compte de la particularité des problèmes semi-infinis convexes. Nous avons commencé par **la nouvelle méthode d'échange** qui utilise le théorème de K.K.T pour la résolution de ses sous-problèmes tout en gardant les contraintes actives avec les multiplicateurs associés sans faire appel à l'optimisation de la contrainte g pour le test d'optimalité. Nous avons ensuite étudié le mécanisme de **la méthode des coupes** pour la résolution du problème semi-infini convexe illustrée par un simple exemple d'application.

Bibliographie

- [1]: LIPING ZHANG, SOON-YI WU , AND MARCO A.LÓPEZ
A new exchange method for convex semi-infinite programming .Année 2010
- [2]: Xavier Antoine, Pierre Dreyfuss et Yannick Privat,ENSMN-ENSEM 2A
Introduction à l'Optimisation : aspects Théoriques, Numériques et Algorithmes.
(2006-2007).
- [3]: Michel Minoux
Programmation Mathématique Théorie et Algorithmes , Collection Technique
et Scientifique des Télécommunications, Volume1. Année 1983
- [4]: M.K.Luhandjula and Ouanes : A cutting-plane approach for semi-infinite mathe-
matical programming - june 2001
- [5]: KARLIN S (1959),Mathematical methods and theory in games,programming
and economies,Vol 1,Addison Wesley mathematical programming
- [6]: SLATER M. (1950), Lagrange Multipliers Revisited: A Contribution to Nonlinear
Programming, Cowles Commission Discussion Paper, Mathematics 403.
- [7]: FIACCO A. V., McCoRMicK G. P. (1968), Nonlinear programming, John Wiley,
New York.
- [8]: KUHN W. W., TUCKER A. W. (1951), Nonlinear programming, Proc, 2nd
Berkeley Symp. on Mathematical Statistics and Probability, University of Cali-
fornia Press, Berkeley.
- [9]: Aude Rondepierre et Sébastien Tordeux ,
Introduction à l'Optimisation Numérique , Année 2009-2010.

- [10]: Sébastien LENGAGNE^{1,2,3}, Nacim RAMDANI^{2,3}, Philippe FRAISSE^{1,3,4}.
a Méthode pour la Planification de Trajectoires Garanties. (Oct 2008)
- [14]: GILL P. E., MURRAY W. (1972), Quasi-Newton Methods for Unconstrained
Optimization, J. Institute of Math. and Its Appl. 9.
- [16]: Introduction à l'Optimisation Différentiable,
a Michel Bierlaire, Presses polytechniques et Universitaires Romandes. Année (2006)
- [18]: Riccardo sacco, Fausto saleri, a Méthodes Numériques, Algorithmes,
Analyse, et Applications edition Springer-Verlag Italia, Milano 2007.

