

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI, TIZI-OUZOU.

FACULTE: DES SCIENCES

DEPARTEMENT : MATHEMATIQUES

MEMOIRE DE MASTER

SPÉCIALITE : RECHERCHE OPÉRATIONNELLE

OPTION : MÉTHODES ET MODÈLES DE DÉCISION

Présenté par:

M^{elle} GATER Naima et *M^{elle}* CHIKHAOUI Manel

Sujet:

Optimisation globale des fonctions deux fois différentiable

Devant le jury d'examen composé de :

| | | | |
|--------------------------|---------------------|---------|----------------|
| M ^r B.OUKACHA | ; Maître de conf | ; UMMTO | ; Président |
| M ^r M.OUANES | ; Maître de conf | ; UMMTO | ; Rapporteur |
| M ^{me} K.LOUADJ | ; Maître assistante | ; UMMTO | ; Examinatrice |
| M ^r K.KASDI | ; Maître assistant | ; UMMTO | ; Examineur |

Soutenu le : 11 / 10 / 2012

Table des matières

| | |
|---|-----------|
| Introduction générale | 4 |
| 1 Notions de base | 6 |
| 1.1 Introduction | 6 |
| 1.2 Formulation des programmes mathématiques et autres définitions pertinentes | 6 |
| 1.3 Rappel de notions d'analyse et d'algèbre | 8 |
| 1.3.1 Matrice | 8 |
| 1.3.2 Critère de Sylvester (mineurs principaux:) | 9 |
| 1.3.3 Un autre critère qui utilise les valeurs propres | 10 |
| 1.3.4 Fonction de plusieurs variables | 10 |
| 1.3.5 Condition d'optimalité pour les extremums d'une fonction | 12 |
| 1.3.6 Cas des fonctions convexes : condition nécessaire et suffisante d'optimalité global | 15 |
| 1.3.7 Optimisation convexe | 15 |
| 1.3.8 Éléments convexes | 15 |
| 1.3.9 Enveloppe convexe : | 16 |
| 1.3.10 Fonction convexe | 17 |
| 1.3.11 Fonction concave | 19 |
| 1.3.12 L'estimation | 20 |
| 1.3.13 Analyse d'intervalle | 21 |
| 1.3.14 Arithmétique d'intervalle: | 21 |
| 1.3.15 Extension naturelle et propriétés | 24 |
| 1.4 La méthode Branch and Bound | 25 |
| 1.4.1 Le principe de la méthode | 25 |
| 1.4.2 L'algorithme de base de la méthode Branch and Bound | 26 |
| 1.4.3 La convergence de la méthode | 27 |
| 2 Optimisation globale des fonctions deux fois différentiable | 30 |
| 2.1 Introduction | 30 |

| | | |
|----------|---|-----------|
| 2.2 | L'optimisation globale des fonctions deux fois différentiable | 31 |
| 2.2.1 | Introduction | 31 |
| 2.2.2 | La sous-estimation des termes bi linéaires: | 31 |
| 2.2.3 | La sous-estimation des termes tri linéaires: | 32 |
| 2.2.4 | La sous-estimation des termes fractionnaires: | 33 |
| 2.2.5 | La sous-estimation des termes tri linéaire fractionnaire: | 33 |
| 2.2.6 | La sous estimation des termes concaves uni- variantes: | 33 |
| 2.2.7 | La sous-estimation des termes non convexes | 34 |
| 2.2.8 | Sous-estimateur convexe: | 35 |
| 2.2.9 | Les contraintes d'égalités: | 36 |
| 2.3 | La construction de la borne inférieure | 37 |
| 2.3.1 | Techniques de calcul de α : | 37 |
| 2.3.2 | Évaluation de méthodes de calcul de α : | 38 |
| 2.3.3 | La méthode du changement uniforme de la diagonale de la matrice Hessienne | 40 |
| 2.3.4 | Méthode basée sur le théorème de Kharitonov | 41 |
| 2.4 | La Méthode de la matrice Hessienne d'intervalle (Hertz's method) | 43 |
| 2.5 | Relaxation de la fonction borne inférieure | 47 |
| 2.6 | Algorithme α BB | 48 |
| 2.6.1 | La convergence de l'algorithme α BB | 49 |
| 2.7 | Conclusion | 50 |
| 3 | Exemples d'applications | 51 |
| 3.0.1 | Introduction | 51 |
| 3.0.2 | Application de la méthode Branch-and-Bound à l'optimisation globale des fonctions deux fois différentiable dans R^n | 51 |
| 3.0.3 | Conclusion | 56 |
| 4 | Programmation dans LINGO | 57 |
| 4.1 | Introduction | 57 |
| 4.1.1 | Présentation du logiciel | 57 |
| 4.1.2 | Interface du logiciel | 58 |
| 4.1.3 | Résolution des exemples traités dans le chapitre 3 sur LINGO avec la méthode Branch-and-Bound | 64 |
| 4.1.4 | Conclusion | 68 |
| | conclusion et perspectives | 69 |

| | |
|---------------------------|----------|
| <u>Table des matières</u> | <u>3</u> |
| Bibliographie | 70 |
| annexe | 72 |

Introduction générale

Nous faisons tous de l'optimisation. Dans notre vie quotidienne, nous cherchons à optimiser notre temps de travail, nos espaces de rangement, ou encore le trajet que nous aurons à parcourir pour nous rendre quelque part, etc. Nous recherchons tous une meilleure solution aux problèmes qui jalonnent notre existence. De manière générale, l'optimisation va donc consister à trouver cette meilleure solution.

Comme nous le rappelle l'adage populaire selon lequel : "les mathématiques permettent de mettre le monde en équation", il peut être tracé un parallèle entre l'optimisation quotidienne et celle, plus technique que l'on retrouve en science. En mathématique, la meilleure solution se recherche au sein d'un domaine initial. Cette solution est souvent soumise à des contraintes qui correspondent à des obligations ou des souhaits à respecter.

L'optimisation mathématique va consister à rechercher dans le domaine initial une solution qui maximise ou minimise une fonction objectif tout en respectant des contraintes. Pour un domaine continu, on distingue classiquement deux types d'optimisation :

L'optimisation locale recherche une solution qui est la meilleure localement, c'est-à-dire que dans son voisinage aucune solution n'est meilleure qu'elle. Cette solution est appelée un optimum local.

L'optimisation globale recherche quant à elle la meilleure solution du domaine en entier, c'est-à-dire que dans tout le domaine il n'existe aucune solution qui lui soit meilleure tout en respectant les contraintes. Cette solution est appelée l'optimum global.

L'intérêt de l'optimisation globale par rapport à l'optimisation locale est patent. Elle garantit en effet que personne ne peut avoir une solution meilleure que celle trouvée. Or, pour une entreprise, cette information a son importance, car la différence entre la solution globale et une solution locale est bien souvent significative. Mais l'intérêt n'est pas que compétitif. Dans de nombreux problèmes, l'optimum global est la seule solution mathématique correspondant à une réalité physique.

De nos jours, afin de résoudre des problèmes d'optimisation globale avec contraintes, de nombreuses stratégies algorithmique s'avèrent disponibles. Pour guider le choix de la meilleure stratégie à utiliser, il est nécessaire de regarder : (i) la taille du problème, (ii) les propriétés de la fonction objectif et les contraintes (continuité, différentiable, linéarité, convexité. . .), (iii) ainsi que le temps disponible pour résoudre le problème. Ces dernières années de nombreux travaux ont été effectués sur les différentes approches (tel que les

méthodes multi-start, les algorithmes révolutionnaires, Les méthodes métaheuristiques et les méthodes déterministes globales).

Nous allons nous intéresser aux méthodes déterministes , parmi elles la méthode Branch-and-Bound qui nous offre la certitude d'obtenir l'optimum global (si celui-ci existe). Ce type d'algorithme dit de séparation et évaluation qui est plus connue sous le nom Branch and Bound qui peut théoriquement résoudre n'importe quel problème.

Notre étude se base sur les problèmes d'optimisations globales des fonctions deux fois différentiables.

Le premier chapitre de notre étude commence par un rappel des notions élémentaires de la programmation mathématique ainsi que les notions et définitions qui en découlent afin d'établir clairement les bases sur lesquelles repose notre approche .

Nous y détaillerons le principe de l'algorithme de base Branch-and-Bound en optimisation globale .

Le deuxième chapitre est consacré quant à lui à l'étude des fonctions deux fois différentiables en sous-estimant ces fonctions selon leurs natures . Une nouvelle technique de calcul de ces sous-estimateurs basée sur un paramètre α .

La résolution de ce type de problème est faite par l'algorithme α BB .

L'application de cet algorithme pour la résolution des problèmes d'optimisations globales des fonctions deux fois différentiables dans \mathcal{R}^n sera présentée dans le troisième chapitre. Des exemples numériques seront traités afin de montrer l'efficacité de cet algorithme, avec une constante α calculée.

Le dernier chapitre sera consacré pour résoudre les exemples traités dans le chapitre 3 en utilisant le logiciel LINGO.

Quelques notions de base seront données sur ce logiciel au début de ce chapitre, les représentations graphiques sont faites en utilisant le logiciel MATLAB 7.

Nous terminerons notre travail par une conclusion générale et des perspectives .

Chapitre 1

Notions de base

1.1 Introduction

Nous proposons dans ce premier chapitre introductif les notions de base sur l'étude de l'optimisation et les notions d'analyse d'intervalles .

On parlera aussi d'une technique principale fréquente utilisée dans l'optimisation globale qui est la méthode de séparation et évaluation (Branch and Bound).afin d'établir clairement les bases sur lesquelles repose notre approche.

1.2 Formulation des programmes mathématiques et autres définitions pertinentes

La programmation mathématique porte sur l'optimisation d'une fonction à plusieurs variables liées par un ensemble d'équations ou d'inequations appelées contraintes. On appelle fonction coût ou fonction critère la fonction à minimiser ou à maximiser et on nomme respectivement contraintes d'égalité et contraintes d'inégalités les équations et les inequations .

Lorsque toutes ces fonctions sont linéaire on parle de la programmation linéaire ,dans le cas contraire ,on parle de la programmation non linéaire. De façon générale, un programme non linéaire s'écrit selon la forme standard suivante :

$$\begin{cases} \text{Min} f(x) \\ g_i(x) \leq 0 & i = 1, \dots, m \\ h_j(x) = 0 & j = 1, \dots, p \\ X \in R^n, \end{cases}$$

Où f, g, h de classe C^2, X un vecteur de taille n . Presque tous les programmes mathématiques

peuvent être mis à la forme standard par quelques transformations élémentaires. Ainsi, un problème de maximisation se transforme en problème de minimisation par l'égalité :

$$\max f(x) = -\min(-f(x)).$$

Une contrainte d'inégalité dans le mauvais sens qui comporte un second membre non nul, soit :

$$g_i(x) \geq b_i$$

Peut s'écrire sous la forme:

$$-g_i(x) + b_i \leq 0.$$

Une contrainte d'inégalité:

$$g_i(x) \leq 0.$$

Peut être transformée en contrainte d'égalité par l'ajout d'une variable d'écart non négative selon l'une ou l'autre formulation suivante :

$$g_i(x) + x_{n+i} = 0; \quad x_{n+i} \geq 0$$

On transforme une contrainte d'égalité en deux contraintes d'inégalité, par exemple :

$$h_j(x) = 0 \Leftrightarrow \begin{cases} h_j(x) \leq 0 \\ -h_j(x) \leq 0 \end{cases}$$

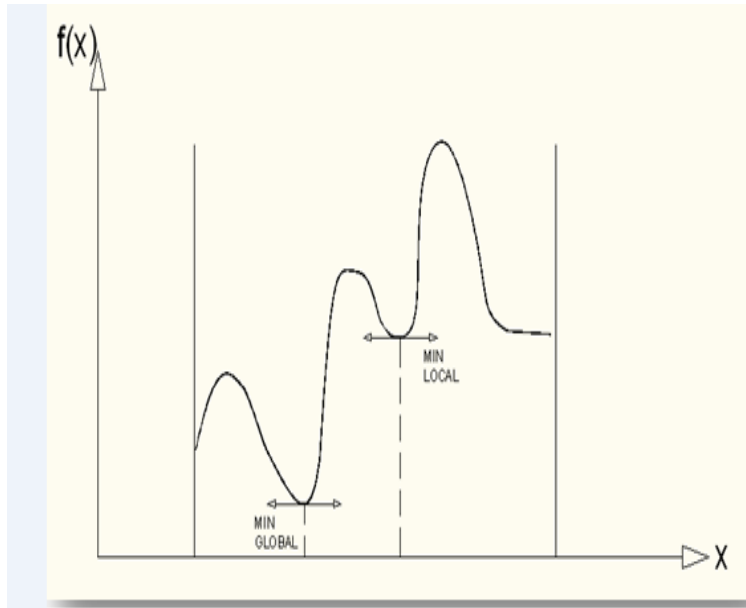
Définition 1.2.1. [5]

Un minimum global est un point admissible x^* qui est selon la fonction critère meilleur ou aussi bon que n'importe quel autre point réalisable c'est-à-dire :

$$f(x^*) \leq f(x) \forall x \in S$$

Ce minimum est seulement local si l'inégalité précédente n'est valable que dans un voisinage immédiat de x^* c'est-à-dire :

$$f(x^*) \leq f(x) \quad \forall x \in v(x^*)$$

FIG. 1.1 – *Optimum local et optimum global*

1.3 Rappel de notions d'analyse et d'algèbre

1.3.1 Matrice

Définition 1.3.1.

On définit une matrice carrée $A(n \times n)$ par un tableau:

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$$

Avec $a_{ij} \in \mathbb{R}; i = 1 \dots n; j = 1 \dots n$.

On a n lignes et n colonnes.

Le vecteur (a_{i1}, \dots, a_{in}) est appelé i^{eme} ligne.

Le vecteur $\begin{pmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{pmatrix}$ est appelé j^{eme} colonne.

Définition 1.3.2.

Soit A une matrice carrée symétrique :

1/ on dit que A est définie positive si :

$$X^T A X > 0$$

$$\forall X \neq 0.$$

2/ on dit que A est semi-définie positive si :

$$X^T A X \geq 0$$

$$\forall X \neq 0.$$

3/ A est définie négative si :

$$X^T A X < 0$$

$$\forall X \neq 0.$$

4/ A est définie semi-négative si :

$$X^T A X \leq 0$$

$$\forall X \neq 0.$$

1.3.2 Critère de Sylvester (mineurs principaux:)

Soit A une matrice carré-symétrique ($n \times n$)

1/ A est définie positive si : $\det A_K > 0 \quad \forall K = 1, \dots, n.$

2/ A est définie négative si : $(-1)^K \det A_K > 0 \quad \forall K = 1, \dots, n.$

$$A_K = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix}, A = \begin{pmatrix} a_{11} & \dots & a_{1K} & \dots & a_{1n} \\ a_{K1} & \dots & a_{KK} & \dots & a_{Kn} \\ a_{n1} & \dots & a_{nK} & \dots & a_{nn} \end{pmatrix}$$

Remarque 1.1.

$\det A_K$ sont appelés mineurs principaux de la matrice A ; $K = 1, \dots, n$ [9],[12].

1.3.3 Un autre critère qui utilise les valeurs propres

A matrice carrée symétrique ; soit λ_i les valeurs propres de A .

- 1/ Si $\lambda_i > 0, \forall i \Rightarrow A$ est définie positive.
- 2/ Si $\lambda_i \geq 0, \forall i \Rightarrow A$ est semi-définie positive.
- 3/ Si $\lambda_i < 0, \forall i \Rightarrow A$ est définie négative.
- 4/ Si $\lambda_i \leq 0, \forall i \Rightarrow A$ est semi-définie négative.

1.3.4 Fonction de plusieurs variables

Définition 1.3.3.

Soit D un domaine de R^n , une application de D à valeurs dans R est dite fonction réelle de n variables :

$$f : D \subseteq R^n \rightarrow R$$

$$p = (x_1, \dots, x_n) \rightarrow f(x_1, \dots, x_n)$$

Définition 1.3.4. [12]

Soit f une fonction définie comme suit :

$$f : D \subseteq R^n \rightarrow R$$

f est continue en P_0 tel que $P_0 \in D$ si : $\forall \epsilon > 0, \exists \delta > 0$ tel que :

$$\|P - P_0\| < \delta \Rightarrow |f(P) - f(P_0)| < \epsilon.$$

On écrit : $\lim_{P \rightarrow P_0} f(P) = f(P_0)$

Définition 1.3.5.

On dit que f est continue sur D si elle est continue en chaque point de D .

Définition 1.3.6.

On définit la dérivée partielle de f par rapport à x_i au point $P = (x_1, \dots, x_n)$ par la limite suivante :

$$\lim_{h \rightarrow 0} \frac{f(x_1, \dots, x_i + h, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{h} \text{ quand elle existe et on la note : } \frac{\delta f}{\delta x_i}(p)$$

Définition 1.3.7.

On définit le gradient de f par :

$$\nabla f(p) = \left(\frac{\delta f}{\delta x_1}(p), \dots, \frac{\delta f}{\delta x_n}(p) \right)$$

Définition 1.3.8.

On dit que f est continûment différentiable sur D (classe C^1) si ses dérivées partielles sont continues sur D .

Exemple 1.1.

$$f(x_1, \dots, x_n) = \sum_{i=1}^n x_i^2$$

$$\frac{\delta f}{\delta x_i}(x_1, \dots, x_n) = 2x_i \quad \forall i \text{ continue sur } R^n \Rightarrow f \text{ est de classe } C^1.$$

Définition 1.3.9.

Soit $F : D \subset R^n \rightarrow R^n$ on définit la matrice Jacobienne de F par:

$$JF(x) = \begin{pmatrix} \Delta f_1(x) \\ \vdots \\ \Delta f_n(x) \end{pmatrix} = \begin{pmatrix} \frac{\Delta f_1}{\delta x_1} & \cdots & \frac{\Delta f_1}{\delta x_n} \\ \vdots & \ddots & \vdots \\ \frac{\Delta f_n}{\delta x_1} & \cdots & \frac{\Delta f_n}{\delta x_n} \end{pmatrix}$$

OU $X = (x_1, \dots, x_n)$; $JF(x)$ ($n \times n$) matrice.

Exemple 1.2.

Écrire la matrice Jacobienne de F tel que :

$$F : R^2 \rightarrow R^2$$

$$(x, y) \rightarrow F(x, y) = (2x + y, 3y)$$

$$JF(x) = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}$$

Définition 1.3.10.

Si les dérivées partielles de f existent et sont derivable sur D , leurs dérivées partielles sont pour f les dérivées partielles secondes et on écrit:

$$\frac{\delta}{\delta x_j} \left(\frac{\delta f}{\delta x_i} (x) \right) = \frac{\delta^2 f}{\delta x_i \delta x_j} (x)$$

Quand : $i = j$: $\frac{\delta^2 f}{\delta x_i^2} (x)$ sont appelées dérivées seconde .

~ $i \neq j$: $\frac{\delta^2 f}{\delta x_i \delta x_j} (x)$ sont appelées dérivées mixtes.

Définition 1.3.11.

Soit f une fonction de classe C^2 (deux fois différentiable) c'est -à-dire les dérivées secondes et mixtes sont continues; on définit la Hessienne de f la matrice Jacobienne du gradient de f notée:

$$\nabla f : R^n \rightarrow R^n$$

$$X \rightarrow \left(\frac{\delta f}{\delta x_1}, \dots, \frac{\delta f}{\delta x_n} \right)$$

$$Hf(x) = J(\nabla f(x)) = \begin{pmatrix} \nabla \frac{\delta f}{\delta x_1}(X) \\ \vdots \\ \nabla \frac{\delta f}{\delta x_n}(X) \end{pmatrix} = \begin{pmatrix} \frac{\delta}{\delta x_1}(\frac{\delta f}{\delta x_1}(X)) & \dots & \frac{\delta}{\delta x_n}(\frac{\delta f}{\delta x_n}(X)) \\ \vdots & \ddots & \vdots \\ \frac{\delta}{\delta x_1}(\frac{\delta f}{\delta x_n}(X)) & \dots & \frac{\delta}{\delta x_n}(\frac{\delta f}{\delta x_n}(X)) \end{pmatrix}$$

Remarque 1.2.

$Hf(x)$ est une matrice carrée ($n \times n$) symétrique.

Exemple 1.3.

Écrire la Hessienne de: $f(x,y,z) = x^2 + y^2 + z^2 + 2xy + 2yz + 2zx$

$$Hf = J(\nabla f)$$

$$Hf = J(2x + 2y + 2z, 2y + 2x + 2z, 2z + 2y + 2x)$$

$$Hf = \begin{pmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{pmatrix}$$

1.3.5 Condition d'optimalité pour les extremums d'une fonction

Chercher un optimum est une chose, mais faut-il avoir un moyen de le reconnaître qui soit plus pratique que celui de le comparer avec la multitude des autres points voisins.

Les optimums satisfont à des conditions d'optimalité qui permettent de les reconnaître.

Le cas sans contrainte constitue une bonne introduction à ce genre de résultats théoriques bien utiles. D'autre part, dans le cas où les variables x_1, \dots, x_n sont astreintes à vérifier des conditions supplémentaires ($g_i(X) \leq 0 \quad i = 1, \dots, m; \quad h_j(X) = 0 \quad j = 1, \dots, p$) peuvent dans certaines conditions se ramener à la résolution de problèmes d'optimisation sans contrainte.

Condition nécessaire d'optimalité locale [5]

Soit X^* un minimum de la fonction $f(X)$.

pour tout $t > 0$ assez petit et $d \in \mathbb{R}^n$ on écrit la formule de Taylor:

$$\begin{aligned} f(X^* + td) &= f(X^*) + t d \nabla f(X^*) + o(\|td\|) \\ \Rightarrow f(X^* + td) - f(X^*) &= t \nabla f(X^*) d + o(\|td\|) \\ \Rightarrow \frac{f(X^* + td) - f(X^*)}{t} &= \frac{t \nabla f(X^*) d + o(\|td\|)}{t} \end{aligned}$$

D'où :

$$\lim_{t \rightarrow 0^+} \frac{f(X^* + td) - f(X^*)}{t} = \lim_{t \rightarrow 0^+} (t \nabla f(X^*) d + o(\|td\|))$$

En particulier avec: $d = -\nabla f(X^*)$, on obtient:

$$\nabla f(X^*) (-\nabla f(X^*)) \geq 0 \Rightarrow -\|\nabla f(X^*)\|^2 \geq 0$$

$$\Rightarrow -\|\nabla f(X^*)\|^2 = 0$$

$$\Rightarrow \|\nabla f(X^*)\|^2 = 0$$

D'où : $\nabla f(X^*) = 0$.

Ceci a donné l'énoncé d'une première condition d'optimalité dite condition nécessaire du premier ordre :

Un minimum X^* de $f(X)$ sur \mathbb{R}^n annule le gradient de la fonction , c'est-à-dire que :

$$\nabla f(X^*) = 0.$$

Définition 1.3.12. [9]

Les points qui vérifient $\nabla f(X^*) = 0$ sont appelés points critiques ou points stationnaires.

Théorème 1.1.

Une condition nécessaire du deuxième ordre pour que X^ soit un minimum (local ou global) de f est :*

1/ $\nabla f(X^*) = 0$ (stationnarité).

2/ $Hf(X^*)$ est une matrice semi-définie positive.

Preuve.

1/ déjà démontré.

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

$$f(X^* + td) = f(X^*) + \nabla f(X^*)td + \frac{1}{2}(td)^T Hf(X^*)td + o(\|td\|^2)$$

$$\Rightarrow \frac{f(X^*+td)-f(X^*)}{t^2} = \frac{\frac{1}{2}t^2 d^T Hf(X^*)d + o(t^2)\|d\|^2}{t^2}$$

D'où :

$$\lim_{t \rightarrow 0} \frac{f(X^*+td)-f(X^*)}{t^2} = \lim_{t \rightarrow 0} \frac{\frac{1}{2}t^2 d^T Hf(X^*)d + o(t^2)\|d\|^2}{t^2}$$

$$\frac{1}{2}d^T Hf(X^*)d + \lim_{t \rightarrow 0} \frac{o(t^2)}{t^2} \|td\|^2 = \frac{1}{2}d^T Hf(X^*)d \geq 0$$

$Hf(X^*)$ est semi-définie positive.

Remarque 1.3.

la figure ci-dessous illustre le fait que les conditions 1/ et 2/ du théorème ne sont pas suffisantes pour garantir un minimum local ou global. À fortiori la stationnarité seule, bien que nécessaire, n'est pas une condition suffisante d'optimalité locale.

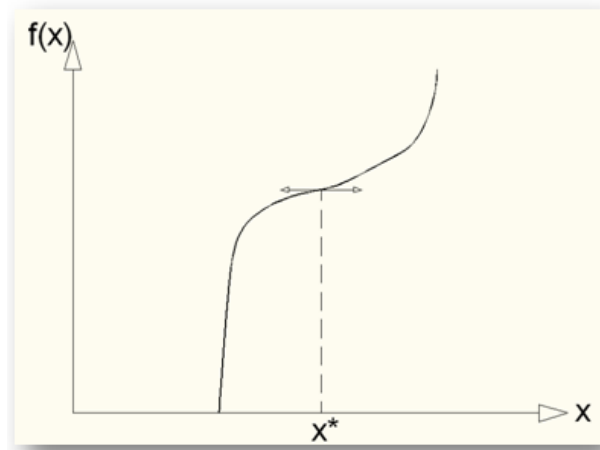


FIG. 1.2 – X^* est un point stationnaire (mais ce n'est pas un minimum local)

Condition suffisante d'optimalité locale

Théorème 1.2. [5]

Une condition suffisante pour que X^* soit un optimum local de f sur R^n est :

1/ $\nabla f(X^*) = 0$ (stationnarité).

2/ $Hf(X^*)$ est une matrice définie positive .

Preuve.

On a le résultat d'algèbre : $\lambda_{min} = \min_{X \neq 0} \frac{X^t A X}{\|X\|^2}$

λ_{min} : La plus petite valeur propre de A

$\lambda_{max} = \max_{X \neq 0} \frac{X^t A X}{\|X\|^2}$

λ_{max} : La plus grande valeur propre de A

On a:

$$\min_{X \neq 0} \frac{X^t A X}{\|X\|^2} \leq \frac{X^t A X}{\|X\|^2} \leq \max_{X \neq 0} \frac{X^t A X}{\|X\|^2}.$$

$$\Rightarrow \lambda_{min} \|X\|^2 \leq X^t A X \leq \lambda_{max} \|X\|^2.$$

Le développement de Taylor de f à l'ordre 2 au niveau de X^* s'écrit alors :

$$f(X^* + d) = f(X^*) + \nabla f(X^*)d + \frac{1}{2}(d)^T Hf(X^*)d + o(\|d\|^2)$$

$$\Rightarrow f(X^* + d) - f(X^*) = \frac{1}{2}(d)^T Hf(X^*)d + o(\|d\|^2) \geq \frac{1}{2}\lambda_{min}\|d\|^2 + o(\|d\|^2)$$

$$\Rightarrow \frac{1}{2}\lambda_{min}\|d\|^2 + o(\|d\|^2) \geq 0$$

$$f(X^* + d) \geq f(X^*)$$

X^* est un minimum local.

1.3.6 Cas des fonctions convexes : condition nécessaire et suffisante d'optimalité global

Dans le cas d'une fonction convexe f définie sur R^n une condition nécessaire et suffisante pour que X^* soit un minimum global de f est que ξ un sous gradient de f en X^* . Pour une fonction continûment différentiable, on obtient donc:

Théorème 1.3.

si f une fonction convexe continûment différentiable une condition nécessaire et suffisante pour que X^ soit un optimum global de f sur R^n est que : $\nabla f(X^*) = 0$.*

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

Remarque 1.4.

On appelle sous gradient de f au point X^0 tout vecteur $Y = (y_1, \dots, y_n)^T \in R^n$ vérifiant : $f(X) \geq f(X^0) + Y^T(X - X^0), \forall X \in R^n$.

1.3.7 Optimisation convexe

Dans les problèmes d'optimisation, avec ou sans contraintes, la convexité joue un rôle très important : celle de convexité .En effet pour la plupart des algorithmes que nous décrirons, la convergence vers un optimum global ne pourra être démontré qu'avec des hypothèses de convexité. Dans ce premier chapitre nous introduisons rapidement quelques éléments de l'analyse convexe requis pour l'étude de l'optimisation.

1.3.8 Éléments convexes

Un ensemble $S \subset R^n$ est dit convexe si et seulement si:

$$\forall x \in S, \forall y \in S \quad \forall \lambda \in [0,1] \Rightarrow \lambda x + (1 - \lambda)y \in S$$

.

C'est-à-dire 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 .

Plus généralement, étant donné p points de $R^n(x^1, \dots, x^p)$ ont dit que $x \in R^n$ est combinaison convexe de ces points s'ils existent des coefficients $U_1, \dots, U_p, U_i \geq 0, \forall i = 1, 2, \dots, p$ tel que

$$\sum_{i=1}^p U_i = 1 \text{ et } \sum_{i=1}^p U_i x^i = x$$

On vérifie aisément qu'un ensemble $S \subset \mathbb{R}^n$ est convexe si et seulement si toute combinaison convexe de points de S est dans S [12].



FIG. 1.3 –

1.3.9 Enveloppe convexe :

Étant donné $S \subset \mathbb{R}^n$, on notera $conv(S)$ l'enveloppe convexe de S c'est-à-dire l'ensemble des points de \mathbb{R}^n qui sont combinaison convexe de points de S . Il découle de ce qui précède que S est convexe si et seulement si $S = conv(S)$.

Exemple 1.4.

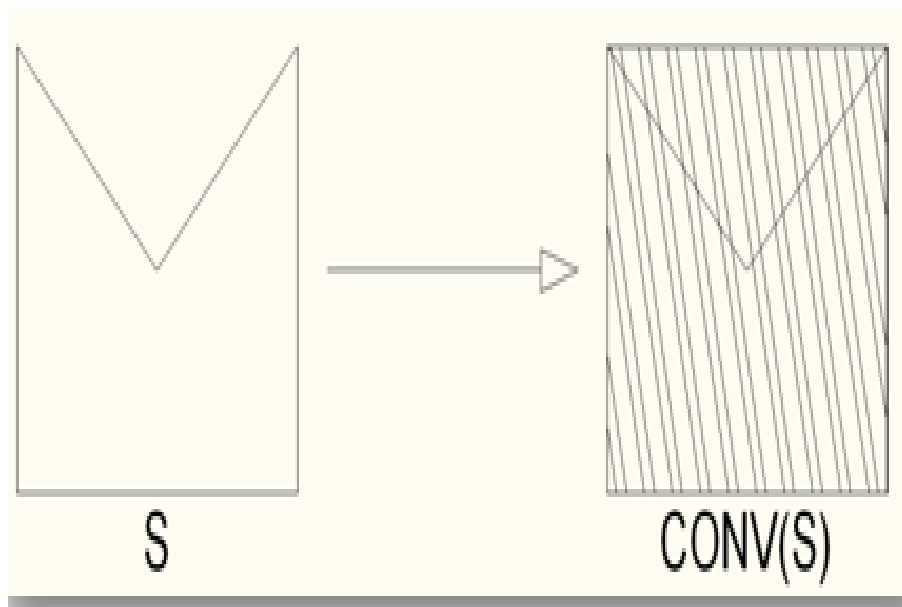


FIG. 1.4 –

1.3.10 Fonction convexe

Une fonction $f : R^n \rightarrow R$ est dite convexe si elle vérifie :

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

Cette inégalité est appelée inégalité de convexité. Elle signifie que toute corde joignant deux points quelconque $(x_1, f(x_1))$ et $(x_2, f(x_2))$ se trouve toujours au dessus de la courbe de f .

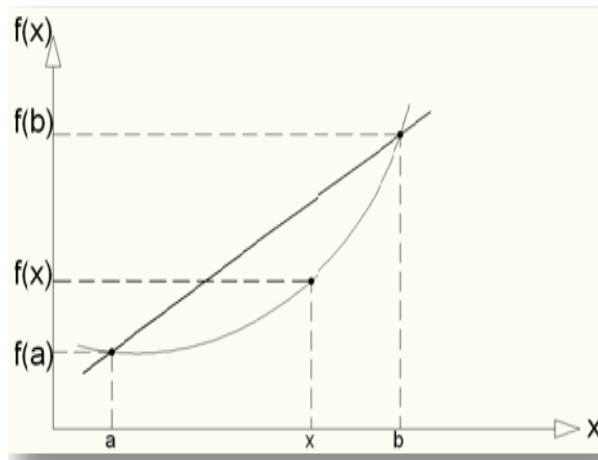


FIG. 1.5 – courbe représentative d'une fonction convexe

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.

Théorème 1.4.

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) (3) ci-dessous sont équivalentes :

1/ f est convexe.

2/ $\forall x \in R^n, \forall x_0 \in R^n, f(x) \geq f(x_0) + \nabla f^T(x_0).(x - x_0)$ c'est-à-dire la courbe C de f est au dessus de ses tangentes.

3/ $\forall x$, la hessienne $\nabla^2 f(x)$ est une matrice semi-définie positive c'est-à-dire :

$$\forall y, y^T \nabla^2 f(x).y \geq 0 \quad [5]$$

Remarque 1.5.

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

f est dite fortement convexe ,s'il existe $\alpha > 0$ tel que $\forall x \in R^n , \forall y \in R^n, \forall \lambda \in [0,1]$

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y) - \frac{1}{2}\alpha\lambda(1 - \lambda) \| x - y \|^2 .$$

On dit aussi que f est α convexe.

1.3.11 Fonction concave

Une fonction $f : R^n \rightarrow R$ est dite concave si $(-f)$ est une fonction convexe i.e avec les mêmes notations , si $\forall x \in R^n , \forall y \in R^n, \forall \lambda \in [0,1], f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$.

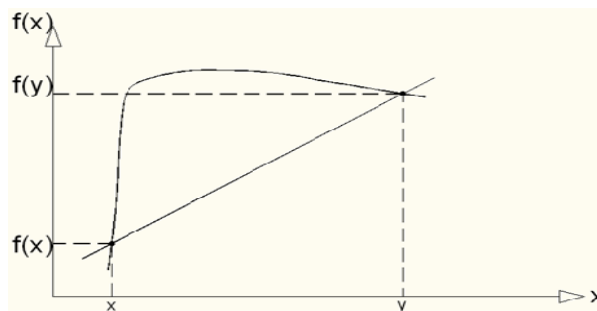


FIG. 1.6 – courbe représentative d'une fonction concave

Remarque 1.6. -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 de minimisation on utilise la notion de convexité.

-Noter que la convexité et la concavité ne sont pas des propriétés complémentaires

*Une fonction peut n'être ni convexe ni concave.

*les fonctions linéaires sont convexes et concaves au même temps .

1.3.12 L'estimation

C'est la connaissance des grandeurs initiales disponibles pour la résolution d'un problème d'extremum est incomplète ,pourtant quoi qu'il en soit la position du problème nous fournit quelques informations sur l'aspect physique et des renseignements correspondants sur les estimations de la précision des données d'entrée appliquées aux problèmes d'extremum tout en étudiant la rapidité de leur convergence.

Disons que les erreurs de calcul sont admissibles si elle ne dépasse pas un certain nombre ε .S'il existe dans une méthode des estimations qui définissent l'ordre de convergence.

Remarque 1.7.

Sous-estimateur : c'est une fonction de borne inférieure.

1.3.13 Analyse d'intervalle**Définition 1.3.13.** [10],[11]

L'analyse d'intervalles est née dans les années 1960 avec l'événement des premiers ordinateurs. Les formats pour coder les nombres flottants n'étaient pas encore standardisés et des erreurs numériques pouvaient s'accumuler et aboutir à des résultats aberrants.

L'analyse d'intervalles a donc été développée à ses débuts pour contourner ce problème.

le premier livre sur ce domaine est celui de Moore en 1966 .Ce n'est pas historiquement le premier travail sur ce sujet. Mais c'est à la suite de ce livre que plus de 1000 articles furent écrit .Ce qui donna un formidable élan à l'analyse d'intervalles.

Exemple 1.5.

Considérons l'équation suivante : $f(x,y) = 33.75y^6 + x^2(11x^2y^2 - y^6 - 121y^4 - 2) + 5.5y^8 + \frac{x}{2y}$.

En l'évaluant au point $x = 77617$ et $y = 33096$, en Fortran double précision, on obtient : $f(77617,33096) = 1.1726039400531\dots$

Alors que le vrai résultat est : $f(77617,33096) = -54767/66192 = -0.8273960599$. Comme on peut le constater, le vrai résultat n'a aucun rapport avec la valeur retournée par l'ordinateur, même le signe n'est pas correct.

Cet exemple montre qu'aujourd'hui encore, malgré la standardisation IEEE-754 du codage des nombres flottants, on n'est jamais à l'abri des erreurs d'arrondi numérique. R étant infini et non dénombrable, il est impossible de représenter exactement tous les nombres.

L'idée de l'analyse d'intervalles est de représenter tous les nombres réels par deux nombres flottants qui l'encadrent.

1.3.14 Arithmétique d'intervalle:

Définissons un intervalle X comme étant une paire ordonnée de nombres réels $[\underline{x}, \bar{x}]$, avec $\underline{x} \leq \bar{x}$. \underline{x} représente la borne inférieure de X et \bar{x} sa borne supérieure.

$$X = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} / \underline{x} \leq x \leq \bar{x}\}.$$

En pratique l'arithmétique d'intervalles est utilisée pour calculer des minorants et des majorants de fonction de R^n dans R .

Notons I l'ensemble des intervalles tel que : $I = \{X = [\underline{x}, \bar{x}] / \underline{x}, \bar{x} \in R \text{ et } \underline{x} \leq \bar{x}\}$.

Soit X et Y deux intervalles appartenant à I . Nous allons maintenant redéfinir dans I les opérateurs usuels de l'arithmétique euclidienne. La plupart des opérateurs et des relations gardent leur définition usuelle sur I , en revanche leurs propriétés sont souvent affaiblies.

Par exemple, la relation d'ordre inférieure ($<$) est une relation d'ordre partiel sur I $X < Y$ si et seulement si $\bar{x} < \underline{y}$.

Les opérateurs sur les ensembles gardent la même définition pour des intervalles. Ainsi, l'union et l'intersection de deux intervalles donnent :

$$X \cup Y = \{\min(\underline{x}, \underline{y}), \max(\bar{x}, \bar{y})\}$$

$$X \cap Y = \begin{cases} \emptyset, & \text{si } \bar{x} < \underline{y} \text{ ou } \bar{y} < \underline{x} \\ [\max(\underline{x}, \underline{y}), \min(\bar{x}, \bar{y})], & \text{sinon} \end{cases}$$

Définissons $w(x)$ comme étant la largeur de l'intervalle X , $w(x) = \bar{x} - \underline{x}$

Si $X = (x_1, x_2, \dots, x_n) \in I^n$, alors $w(x) = \max_{i \in \{1, \dots, n\}} w(x_i)$.

Notons $mid(x)$, le milieu de l'intervalle X :

$$mid(x) = \frac{\bar{x} + \underline{x}}{2}.$$

Si $X = (x_1, x_2, \dots, x_n) \in I^n$, alors $mid(X) = (\frac{x_1 + \bar{x}_1}{2}, \frac{x_2 + \bar{x}_2}{2}, \dots, \frac{x_n + \bar{x}_n}{2})$.

Les opérateurs usuels de l'arithmétique d'intervalles sont définis de telle sorte que l'intervalle résultant soit le plus petit intervalle contenant tous les points images des éléments des intervalles de départ. Ainsi, la formule générale des opérateurs unaires et binaires devient :

$$X \text{ op } Y = \{x \text{ op } y \mid x \in X \text{ et } y \in Y\} \text{ avec } \text{op} \in \{+, -, \times, \div\}.$$

$U(x) = \{u(x) \mid x \in X\}$ avec $u \in \{\sqrt{\quad}, \log, \exp, \text{abs}, \dots\}$. On obtient les formules suivantes

pour les opérateurs binaires de base:

$$\begin{cases} x + y = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]; x - y = [\underline{x} - \bar{y}, \bar{x} - \underline{y}], \\ x \times y = [\min\{\underline{x} \times \underline{y}, \underline{x} \times \bar{y}, \bar{x} \times \underline{y}, \bar{x} \times \bar{y}\}, \max\{\underline{x} \times \underline{y}, \underline{x} \times \bar{y}, \bar{x} \times \underline{y}, \bar{x} \times \bar{y}\}], \\ x \div y = x \times [\frac{1}{\bar{y}}, \frac{1}{\underline{y}}] \text{ si } 0 \notin y, \end{cases}$$

Il est aussi très simple d'étendre ce principe aux fonctions unaires. Voici quelques exemples

d'opérateurs usuels:

$$\sqrt{x} = [\sqrt{\underline{x}}, \sqrt{\bar{x}}]; \text{ pour } \underline{x} \geq 0.$$

$$\log(x) = [\log(\underline{x}), \log(\bar{x})]; \text{ pour } \underline{x} \geq 0.$$

$$\exp(x) = [\exp(\underline{x}), \exp(\bar{x})].$$

$$\text{abs}(x) = \begin{cases} [\text{abs}(\underline{x}), \text{abs}(\bar{x})], & \text{si } \underline{x} > 0; \\ [\text{abs}(\bar{x}), \text{abs}(\underline{x})], & \text{si } \bar{x} < 0; \\ [0, \max(\text{abs}(\underline{x}), \text{abs}(\bar{x}))], & \text{si } 0 \in x. \end{cases}$$

De nombreuses extensions furent proposées notamment pour inclure les intervalles infinis, ou par exemple pour gérer la division par un intervalle contenant 0, cette arithmétique appelée l'arithmétique d'intervalles étendue a été découverte par Kahan et Hanson de façon indépendante et popularisée par Hansen.

Concernant l'implémentation de l'arithmétique d'intervalles sur machine, il est bien entendu impossible d'utiliser des nombres réels, une version n'utilisant que des nombres flottants est donc nécessaire. Cette arithmétique s'appelle l'arithmétique d'intervalles arrondie. Elle consiste à approximer les nombres réels par Le nombre flottant le plus proche de telle sorte que l'intervalle réel soit inclus dans l'intervalle arrondi[10].

Soit $X \in I$; alors \underline{x} sera arrondi au nombre flottant inférieur le plus proche et \bar{x} sera arrondi au nombre flottant supérieur le plus proche. Une étude pour créer le standard IEEE-P1788 définissant cette arithmétique est actuellement en cours . Mais, il existe déjà de nombreuses implémentations performantes de cette arithmétique, dont voici une liste non exhaustive dans différents langages de programmation [11].

-INTLIB en Fortran 77 et Fortran 90.

-INTLAB en Matlab.

-PROFIL/BIAS en C.

une librairie directement incluse dans le compilateur SUN f90.

Ces différentes librairies permettent d'utiliser le type d'intervalle et de le manipuler comme tous les autres types. Les opérateurs et les fonctions usuelles sont simplement redéfinis par surcharge d'opérateur [11].

1.3.15 Extension naturelle et propriétés

Il existe des définitions plus précises d'une fonction explicite (factorable fonctions en anglais) et d'un problème explicite (factorable program en anglais). Mais, par souci de concision, nous avons préféré utiliser des définitions plus intuitives.

Définition 1.3.14.

Soit $f : R^n \rightarrow R$, f possède une expression explicite si l'expression analytique de f est connue de façon explicite, c'est-à-dire qu'elle peut s'écrire en n'utilisant que des variables, des fonctions et des opérateurs élémentaires tels que $+$, $-$, \times , $/$, $\sqrt{\quad}$, \log , \exp , abs , \cos , \sin , \arccos . On dit que f est une fonction explicite.

Définition 1.3.15.

Un problème explicite est un problème dans lequel toutes les fonctions possèdent des expressions explicites et les contraintes n'utilisent que les relations standards (\leq , $<$, \geq , $>$, $=$).

Définition 1.3.16.

Soit $f : X \subset R^n \rightarrow R$. Une extension naturelle aux intervalles d'une fonction est la réécriture de f en remplaçant toutes les occurrences d'une variable par l'intervalle correspondant et les opérateurs classiques par leur équivalent en arithmétique d'intervalles, elle est notée $EN_f(x)$.

Toutes les fonctions explicites possèdent une extension naturelle. Mais, l'évaluation de celle-ci n'est pas unique. En effet, certaines fonctions peuvent s'écrire sous forme développée ou factorisée.

Il existe donc une extension naturelle pour chaque façon d'écrire la fonction.

Bien entendu, les fonctions explicites ne sont pas les seules à posséder une fonction d'inclusion. Il est par exemple possible d'écrire des fonctions d'inclusion de fonction comportant des `if..then...else`.

Proposition 1.1.

L'extension naturelle aux intervalles d'une fonction est une fonction d'inclusion [10].

Exemple 1.6.

Voici un exemple d'application de l'extension naturelle.

Les occurrences de x_1 sont remplacées par $[1,2]$ et les occurrences de x_2 par $[2,6]$.

Les calculs sont ensuite effectués en utilisant les formules de l'arithmétique d'intervalles.

$$\forall x \in X = [1,2] \times [2,6], f(x) = x_1 \times x_2^2 - \exp(x_1 + x_2).$$

$$EN_f(x) = [1,2] \times [2,6]^2 - \exp([1,2] + [2,6]).$$

$$EN_f(x) = [-2976.9579870417284, 51.91446307681234].$$

Ainsi, -2976.9579870417284 est un minorant et 51.91446307681234 un majorant de f sur $[1,2] \times [2,6]$.

L'analyse d'intervalles peut donc se munir de tous les outils de l'arithmétique classique pour calculer des minorants et des majorants d'une fonction explicite sur un intervalle. Malheureusement, les définitions naturelles proposées n'offrent pas les propriétés d'inversibilité et de distributivité. Notons que l'inverse de l'addition dans I n'est pas la soustraction et l'inverse de la multiplication n'est pas la division. De plus, l'ensemble I muni de la loi \times est sous-distributive. Ainsi, $\forall x, y, z \in I, z \times (x + y) \subseteq z \times x + z \times y$.

Définition 1.3.17. La matrice Hessienne $H_{(f(x))}$ de la fonction $f(x)$ deux fois différentiable est définie sur X peut être transformée en une matrice d'intervalle $H_{(f(x))}$. De même cas pour les polynômes caractéristiques l'ensemble de matrices décrit par $H_{(f(x))}$ contient ceux-là définis par $H_{(f(x))}$

$$H_{(f(x))} = \begin{pmatrix} a_{11}(x) & \dots & a_{1n}(x) \\ \vdots & \ddots & \vdots \\ a_{n1}(x) & \dots & a_{nn}(x) \end{pmatrix} \subseteq H_{(f(x))} = \begin{pmatrix} [a_{11}^L, a_{11}^U] & \dots & [a_{1n}^L, a_{1n}^U] \\ \vdots & \ddots & \vdots \\ [a_{n1}^L, a_{n1}^U] & \dots & [a_{nn}^L, a_{nn}^U] \end{pmatrix}$$

voir [1].

1.4 La méthode Branch and Bound

La méthode Branch and Bound est une méthode qui divise le problème non linéaire donné en plusieurs sous problèmes et qui exploite les informations qu'elle produit durant la résolution de chacun de ces sous problèmes.

De nombreuses améliorations ont été proposées pour cette méthode notamment à partir de différentes structures des problèmes à résoudre, et ça en prenant appui sur les outils classiques de la recherche opérationnelle.

1.4.1 Le principe de la méthode

Soit (p) le problème d'optimisation globale

$$(p) \begin{cases} \min f(x) \\ x \in H \end{cases}$$

Où : $H \subseteq R^n$.

$f : K \rightarrow R$ ($H \subseteq K \subseteq R^n$) f continue et non convexe .

L'algorithme Branch and Bound consiste à engendrer deux suites convergentes $\{UB_K\}$ et $\{LB_K\}$ des bornes supérieure et inférieure respectivement de la valeur minimale de la fonction du problème (p).

UB:Upper bound.

LB:Lower bound.

Une relaxation initial R de l'ensemble réalisable H sera définie telle que: $H \subset R$.

R est convexe ,il peut être un simplex,un rectangle ,un cône,...

A chaque itération K les problèmes de bornes inférieure et supérieure seront résolus sur un nombre fini de sous ensembles de R . On notera ces sous ensembles $R_{K_i} \in I_k$ où I_k est l'ensemble des sous ensembles actifs à l'itération K . Sur chaque sous ensemble R_{K_i} les bornes inférieures et supérieures LB_K et UB_K seront calculées par la relaxation de f sur R_{K_i} et la relaxation de $\min f$ localement sur le sous ensemble réalisable $R_{K_i} \cap H$.

Cette méthode utilise la stratégie "le meilleur d'abord".

En effet ,les bornes inférieures supérieures finales pour l'itération K seront données par:

$$\begin{cases} LB_K = \min LB_K \\ UB_K = \min UB_K \end{cases}$$

respectivement ,et tout sous ensemble sur lequel la borne inférieure dépasse UB_K sera éliminée, car $\min f$ ne peut être atteint sur un tel sous ensemble .

Au fait ,cette méthode peut se représenter schématiquement par une arborescence qui a pour racine l'ensemble R ,et pour sommet des sous ensembles R_{K_i} qui s'obtiennent par les subdivisions successives ,et deux sommets seront reliés si et seulement si le deuxième sous ensemble est obtenu par la partition directe du premier,et à chaque niveau de l'arborescence créé la borne inférieure et supérieure seront obtenues par l'application d'une recherche locale.voir[2,4,5]

Notons x^* la solution optimal du problème (p) pour ce qui suit .

1.4.2 L'algorithme de base de la méthode Branch and Bound

On peut résumer la procedure par les étapes suivantes:

Algorithme général:

- 1- Construire l'ensemble R tel que : $H \subset R$.
- 2- Posons : $K = 1, I_K = R$, fixer $\epsilon > 0$.

3- Construire les problèmes des bornes inférieure et supérieure de $\min f(x)$ sur R .

Soient LB_K, UB_K des solutions obtenues respectivement.

4- Si: $UB_K - LB_K \leq \epsilon$ donc on s'arrête et on pose:

$\min f(x) = UB_K$ et $x^* = x^K \in \{x : f(x) = UB_K, x \in H \cap R\}$.

5- Sinon: subdiviser I_K en deux sous ensembles (ou en un nombre fini de sous ensembles)

R_{K_1} et R_{K_2} tel que:

$$\bigsqcup_{i=1}^{i=2} R_{K_i} = R \text{ et } \dot{R}_{K_1} \cap \dot{R}_{K_2} = \emptyset.$$

Où: \dot{R} est l'intérieur de R .

6- construire les problèmes des bornes inférieures et supérieures de $\min f(x)$ sur $H \cap R_{K_i}$, $i=1,2$. Soient :

$$LB_{K_1}, UB_{K_1} \text{ et } LB_{K_2}, UB_{K_2}$$

les solutions obtenues.

7- Posons :

$$\begin{cases} UB_{K+1} = \min\{UB_{K_1}, UB_{K_2}, UB_K\} \\ LB_{K+1} = \min\{LB_{K_1}, LB_{K_2}\} = LB_{K^*} \end{cases}$$

8- Posons: $I_K = \{R_{K_1}, R_{K_2}\}$.

9- Éliminer de I_K tous sous ensembles $R_{K_j}, j=1,2$ tel que $LB_{K_j} > UB_{K+1}$ où $H \cap R_{K_j} = \emptyset$.

et posons: $I_{K+1} = R_{K_i^*}$.

10- Posons: $K = k + 1$ et revenant à 4-10.

Notation

R_K : le sous ensemble actuel; LB_K : la borne inférieure (à la K^{eme} itération); UB_K : la borne supérieure (à la K^{eme} itération); x^K : la solution trouvée (à la K^{eme} itération).

1.4.3 La convergence de la méthode

Évidemment si l'algorithme précédent se termine à l'itération j , alors x^j est la solution optimale et UB_j est la valeur optimale de la fonction objectif, mais en général on ne peut pas garantir ça, c'est-à-dire le fait de s'arrêter après un nombre fini d'itérations, et si l'algorithme est infini, alors il engendre au moins une suite R_K infinie des sous ensembles des subdivisions successives telle que $R_{K+1} \subset R_K, K \in N$.

Donc on doit montrer que chaque point d'accumulation de la suite des solutions $\{x^K\}$

correspondante est une solution optimale du problème donné.

Le théorème suivant démontre la convergence de l'algorithme de Branch and Bound.

Théorème 1.5.

Si pour chaque suite infinie $\{R_K\}$, $R_{K+1} \subset R_K, K \in N$ des ensembles des partitions successives, les bornes inférieure et supérieure vérifient :

$$\lim_{K \rightarrow \infty} (UB_K - LB_K) = \lim_{K \rightarrow \infty} (UB_K - LB(R_K)) = 0 \quad (1)$$

Alors

$$UB = \lim_{K \rightarrow \infty} UB_K = \lim_{K \rightarrow \infty} f(X_k) = \lim_{K \rightarrow \infty} LB_K = LB \quad (2).$$

Et chaque point d'accumulation X^ de la suite $\{X_K\}$ est une solution optimale de $\min f(X)$, $x \in H$.*

Preuve.

A l'itération K le sous ensemble R_K sera choisi à partir de la règle suivante :

$$LB_{K+1} = \min\{LB_{K_1}, LB_{K_2}\}$$

de l'algorithme ci-dessus, à la fin de l'itération $(K-1)$ et donc :

$$LB_K = LB(R_K)$$

Soit $\{X_K\}$ la suite des solutions optimales engendrées par l'algorithme, comme H est compact, alors $\{X_K\}$ a des points d'accumulations.

Soit X^* un point d'accumulation de la suite $\{X_K\}$, donc il existe une sous suite infinie de $\{X_K\}$ qui converge vers X^* , et comme f est continue alors :

$$\lim_{K \rightarrow \infty} f(X_K) = f(X^*)$$

.

Posons: $f^* = \min\{f(X), X \in H\}$, la suite $\{LB_K\}$ des bornes inférieures est croissante monotone, majorée par f^* , donc la suite $LB = \lim_{K \rightarrow \infty} LB_K$ existe.

D'autre part, la suite $\{UB_K\}$ des bornes supérieures est décroissante, minorée par f^* ,

donc sa limite: $UB = \lim_{K \rightarrow \infty} UB_K$ existe , et on a: $UB_K = f(X_K) \geq f^*$, et ça implique que :

$$LB \leq f^* \leq \lim_{K \rightarrow \infty} f(X_K) = f(X^*) = UB.$$

Et du fait de(1) on peut déduire directement(2) .

Chaque réalisation de l'algorithme Branch and Bound doit donc spécifier :

1. L'ensemble R tel que $H \subset R$.
2. Les procédures qui donneront les bornes inférieures et supérieures sur les sous-ensembles engendrés par l'algorithme.
3. Les subdivisions successives de R En sous-ensembles.

Bien entendu, ça va dépendre de la structure du problème (P).pour cela nous allons étudier chaque cas a part .en se basant sur les trois points précédents, et en montrant la convergence de l'algorithme a chaque fois.

Pour savoir plus sur cette méthode ,regardez les références :[2],[5]

Chapitre 2

Optimisation globale des fonctions deux fois différentiable

2.1 Introduction

L'optimisation globale est une branche des mathématiques en plein expansion depuis quelques années. Ce domaine étudie dans un cadre générale la question de trouver l'optimum (maximum ou minimum) d'une fonction sous diverses contraintes.

L'intérêt de l'étude de cette classe de problème d'optimisation est qu'elle englobe un vaste éventail d'applications réelles [2].

Le retour au premier plan de l'optimisation globale correspond à un besoin industriel .De nombreuses applications que se soit au niveau de la conception ou de l'exploitation, se ramène à la recherche d'optima qui n'entre pas dans le cadre des hypothèses simplificatrices (convexité, différentiability). Pour cela, il existe plusieurs méthodes algorithmiques qui mènent à la solution sûrement, mais la vitesse de convergence varie selon la méthode utilisée[8].

En particulier parmi maintes méthodes proposées pour la résolution de tels problèmes nous exposons dans ce chapitre la méthode d'optimisation globale Branch and Bound(α BB) basée sur l'analyse d'intervalle ce qui lui donne un caractère déterministe .Elle permet donc de trouver avec certitude l'optimum global quel que soit la nature du problème .Cette méthode compte sur les propriétés mathématiques pour obtenir le problème de la borne inférieure. Cependant la construction rigoureuse de la borne inférieure est seulement possible dans les cas où les expressions analytiques explicites pourraient être dérivées pour les valeurs propres ou des bornes sur les valeurs propres des fonctions deux fois différentiable présentées dans le problème[11].

2.2 L'optimisation globale des fonctions deux fois différentiable

2.2.1 Introduction

Cette partie sera consacrée à l'étude du problème (1), qui garantit les propriétés théoriques de l'algorithme α BB (Branch and Bound) qui est conçu pour résoudre ce genre de problèmes de minimisation.

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

Branch-and-Bound est l'approche qui nécessite l'identification de la valeur propre minimale de la matrice hessienne des fonctions convexes. son algorithme basé sur cette technique converge vers la solution désirée. Il est appliqué en général pour les fonctions non convexes deux fois différentiable. Cependant le calcul exact d'une valeur propre minimale d'une fonction générale sur un intervalle donné d'une variable pose des difficultés. La généralisation d'une borne inférieure sur cette valeur propre conserve les garanties théoriques d'optimalité globale en améliorant le problème .

Le sous-estimateur du problème devrait être trouvé pour satisfaire à cette exigence, les fonctions sont décomposées en somme de termes selon la structure mathématique, chaque terme est assigné à plusieurs catégories pré définies. L'avantage de cette méthode est que chaque classe de termes peut être sous-estimée en utilisant une technique connue et efficace. Pour ce but le problème original doit être transformé en un sous problème convexe qui peut être résolu par l'optimisation globale. Dans le fait de construire des sous estimateurs convexes pour la fonction générale, il est d'abord noté que les termes linéaires et convexes n'exigent pas de transformation. L'enveloppe convexe du bi linéaire, tri linéaire, tri linéaire factionnaire, factionnaire et les termes concaves uni variant peut être construite selon les règles simples suivantes[1],[7].

2.2.2 La sous-estimation des termes bi linéaires:

En cas d'un terme bi linéaire xy , Al-Khayyal et Falk (1983) ont montré que la borne inférieure dans le domaine $[x^L, x^U] \times [y^L, y^U]$ est obtenue en présentant une nouvelle variable w_B qui remplace chaque occurrence de xy dans le problème et est satisfaisant la suite au rapport :

$$w_B = \max\{x^L y + y^L x - x^L y^L; x^U y + y^U x - x^U y^U\} \quad (2)$$

Sa borne inférieure peut être étendue et inclus dans le problème de minimisation en ajoutant deux contraintes d'inégalité linéaires,

$$\begin{aligned} w_B &\geq x^L y + y^L x - x^L y^L \\ w_B &\geq x^U y + y^U x - x^U y^U \end{aligned} \quad (3)$$

De Plus, sa borne supérieure peut être imposé sur w à construire une meilleure approximation du problème original (McCormick, 1976). C'est accompli par l'adjonction de deux contraintes linéaires :

$$\begin{aligned} w_B &\leq x^U y + y^L x - x^U y^L \\ w_B &\leq x^L y + y^U x - x^L y^U \end{aligned} \quad (4)$$

2.2.3 La sous-estimation des termes tri linéaires:

Un terme tri linéaire de la forme xyz peut être sous-estimé d'une manière semblable (Maranas et Floudas, 1995). Une nouvelle variable w_T est présentée et limitée par les huit contraintes d'inégalité suivantes :

$$\begin{aligned} w_T &\geq xy^L z^L + x^L y z^L + x^L y^L z - 2x^L y^L z^L \\ w_T &\geq xy^U z^U + x^U y z^L + x^U y^L z - x^U y^L z^U - x^U y^U z^U \\ w_T &\geq xy^L z^L + x^L y^U z + x^L y^U z - x^L y^U z^U - x^L y^L z^L \\ w_T &\geq xy^U z^L + x^U y z^U + x^L y^U z - x^L y^U z^L - x^U y^U z^U \quad (5) \\ w_T &\geq xy^L z^U + x^L y z^L + x^U y^L z - x^U y^L z^U - x^L y^L z^L \\ w_T &\geq xy^L z^U + x^L y z^U + x^U y^U z - x^L y^L z^U - x^U y^U z^U \\ w_T &\geq xy^U z^L + x^U y z^L + x^U y^U z - x^U y^U z^L - x^L y^L z^L \\ w_T &\geq xy^U z^U + x^U y z^U + x^U y^U z - 2x^U y^U z^U \end{aligned}$$

2.2.4 La sous-estimation des termes fractionnaires:

Les termes fractionnaires de la forme $\frac{x}{y}$ sont sous estimés en présentant une nouvelle variable w_F et deux nouvelles contraintes (Maranas et Floudas, 1995) qui dépendent du signe des limites de x .

$$w_F \geq \begin{cases} x^L/y + x/y^U - x^L/y^U, & \text{si } x^L \geq 0; \\ x/y^U - x^L y/y^L y^U + x^L/y^L, & \text{si } x^L \leq 0 \end{cases}$$

$$w_F \geq \begin{cases} x^U/y + x/y^L - x^U/y^L, & \text{si } x^U \geq 0 \\ x/y^L - x^U y/y^L y^U + x^U/y^U, & \text{si } x^U < 0 \end{cases} \quad (6)$$

2.2.5 La sous-estimation des termes tri linéaire fractionnaire:

Pour les termes tri linéaire fractionnaire huit nouvelles contraintes sont exigées (Maranas et Floudas, 1995). xy/z est remplacé par la variable w_{FT} et les contraintes pour $x^L, y^L \geq 0$ et $z^L > 0$ sont donnés par :

$$w_{FT} \geq xy^L/z^U + x^L y/z^U + x^L y^L/z - 2x^L y^L/z^U$$

$$w_{FT} \geq xy^L/z^U + x^L y/z^L + x^L y^U/z - x^L y^U/z^L - x^L y^L/z^U$$

$$w_{FT} \geq xy^U/z^U + x^U y/z^U + x^U y^L/z - x^U y^L/z^U - x^U y^U/z^L$$

$$w_{FT} \geq xy^U/z^U + x^U y/z^L + x^L y^U/z - x^L y^U/z^U - x^U y^U/z^L$$

$$w_{FT} \geq xy^L/z^U + x^L y/z^L + x^U y^L/z - x^U y^L/z^L - x^L y^L/z^U$$

$$w_{FT} \geq xy^U/z^U + x^U y/z^L + x^L y^U/z - x^L y^U/z^U - x^U y^U/z^L$$

$$w_{FT} \geq xy^L/z^U + x^L y/z^L + x^U y^L/z - x^U y^L/z^L - x^L y^L/z^U$$

$$w_{FT} \geq xy^U/z^L + x^U y/z^L + x^U y^U/z - 2x^U y^U/z^L \quad (7)$$

2.2.6 La sous estimation des termes concaves uni- variantes:

Les fonctions uni variantes concaves sont sous estimés par leur linéarisation . Ainsi l'enveloppe convexe de la fonction concave $ut(x)$ sur $[x^L, x^U]$ est la fonction linéaire de x :

$$ut(x^L) + \frac{ut(x^U) - ut(x^L)}{x^U - x^L}(x - x^L) \quad (8)$$

La généralisation d'un sous estimateur convexe pour une fonction concave uni variante n'exige pas l'introduction de variables supplémentaires ou de contraintes.

2.2.7 La sous-estimation des termes non convexes

Pour les problèmes non convexes en générale, une version légèrement modifiée du sous estimateur proposé par Maranas et Floudas (1994b) est utilisée. Une fonction $f(x) \in C^2(R^n)$ est sous estimée sur le domaine entier $[x^L, x^u]$ par la fonction $L(x)$ défini comme:

$$L(x) = f(x) + \sum_{i=1}^n \alpha_i (x_i^L - x_i)(x_i^U - x_i) \quad (9)$$

Où les α_i sont des scalaires positifs.

Comme le terme de somme dans l'équation (9) est négatif sur la région entière $[x^L, x^U]$, $L(x)$ est sous-estimateur de $f(x)$. En outre, comme le terme quadratique est convexe toutes les fonctions non convexes de $f(x)$ peuvent données les valeurs des paramètres α_i , $L(x)$ est donc un sous-estimateur convexe.

$L(x)$ est convexe si et seulement si sa matrice Hessienne $H_L(x)$ est semi-définie positive, une condition de convexité utile est tirée en notant que $H_L(x)$ est rattaché à la matrice Hessienne $H_f(x)$ de $f(x)$ par:

$$H_L(x) = H_f(x) + 2\Delta \quad (10)$$

Où Δ est une matrice diagonale dont les éléments diagonaux sont les α_i .

L'adjonction du terme quadratique à la fonction $f(x)$ montrée dans l'équation (9) correspond à l'introduction d'un changement des éléments diagonaux de sa matrice Hessienne $H_f(x)$.

Le théorème suivant peut alors être utilisé pour garantir que $L(x)$ est effectivement un sous-estimateur convexe.

Théorème 2.1.

$L(x)$ Comme défini dans l'équation (9) est convexe si et seulement si:

$H_L(x) = H_f(x) + 2\Delta = H_f(x) + 2diag(\alpha_i)$ est semi-définie positive pour tout $x \in [x^L, x^U]$. Deux méthodes déterministes ont été conçues pour identifier automatiquement une matrice diagonale appropriée.

Remarque 2.1.

- N'importe quel terme qui n'appartient pas dans une des classes indiquées ci dessus est

étiqueté comme un terme non convexe général.

-la seule condition est que la fonction doit être deux fois différentiable afin que l'ordre de la procédure des sous estimateurs doit être applicable à tout genre de problème non convexe.

2.2.8 Sous-estimateur convexe:

Pour n'importe quelle fonction donnée deux fois différentiable on peut identifier le sous-estimateur convexe .

les fonctions peuvent donc être décomposées comme :

$$f(x) = LT(X) + CT(X) + \sum_{i=1}^{bt} BT_i(x) + \sum_{i=1}^{nt} NT_i(x).$$

Où :LT est le terme linéaire, CT est un terme convexe, BT sont des termes bi linéaires, l'UT sont des termes concaves uni variante et le NT sont des termes non convexes généraux.

bt,ut et nt dénotent le nombre de termes concaves et uni variante et bi linéaires respectivement.

La borne inférieure correspondante à la fonction est :

$$L(x,w) = LT(X) + CT(X) + \sum_{i=1}^{bt} b_i w_{Bi} + \sum_{i=1}^{tt} t_i w_{Ti} + \sum_{i=1}^{ft} f_i w_{Fi} + \sum_{i=1}^{fTT} t_i w_{FTi} +$$

$$\sum_{i=1}^{ut} (UT_i(x^{i,L}) + \frac{UT_i(x^{i,U}) - UT_i(x^{i,L})}{x^{i,U} - x^{i,L}}) + \sum_{i=1}^{nt} (NT_i(x) +$$

$$\sum_{j=1}^n \alpha_{ij} (x_j - x_j^L)(x_j - x_j^U))$$

Où a_{ij} correspond au terme i et à la variable j qui satisfait le Théorème 2.1, les variables w_{Bi} sont définies par les équations (3) et (4). Chaque sous-estimateur trouvé est une fonction du domaine considéré.

Puisque l'algorithme α BB suit l'approche du branchement et evaluation ce domaine est systématiquement réduit à chaque nouveau nœud de l'arbre pour que les fonctions de bornes inférieures puissent être produites par les mises à jour d'équations (3)- (9).

La décomposition de fonctions dans les termes peut améliorer la performance de l'algorithme α BB de deux façons. D'abord, l'utilisation du sous estimateur pour certains types de termes augmente la qualité des bornes inférieures produites pour la solution optimale.

Deuxièmement, la construction de sous estimateur convexe pour un terme non convexe dépend de la dimension du terme à $O(n^2)$ ou plus haut. La séparation pour de grands termes dans les termes impliquant un plus petit nombre de variables s'ensuit donc dans une diminution dans les frais quantitatifs. D'une façon intéressante, la structure de beaucoup de problèmes physiques se prête naturellement à une telle décomposition.

2.2.9 Les contraintes d'égalités:

Pour trouver la borne inférieure du problème non convexe, les sous estimateurs trouvés dans chaque domaine doivent être convexes.

Cela implique que toutes les contraintes d'inégalité doivent être convexes et linéaires et que la taille de l'intervalle réalisable doit être augmentée par rapport à celle du problème non convexe original.

Une des deux stratégies peut être utilisée pour sous estimer une égalité non linéaire selon le type de termes qu'il implique.

La première approche est utilisée pour les égalités qui sont seulement linéaire, bi linéaire, tri linéaire et tri linéaire fractionnaire. Les termes non linéaires sont remplacés par les nouvelles variables qui sont utilisées d'une façon linéaire. L'égalité provenant de la substitution est donc linéaire. L'ensemble des valeurs de ces nouvelles variables peuvent être atteintes selon les termes non linéaires. Ainsi, étant donné l'égalité :

$$LT(x) + \sum_{i=1}^{bt} b_i x_{B_{i,1}} x_{B_{i,2}} + \sum_{i=1}^{tt} t_i x_{T_{i,1}} x_{T_{i,2}} x_{T_{i,3}} + \sum_{i=1}^{ft} f_i \frac{x_{F_{i,1}}}{x_{F_{i,2}}} + \sum_{i=1}^{ftt} f_{t_i} \frac{x_{F_{i,1}} x_{F_{i,2}}}{x_{F_{i,3}}} = 0 \quad (11)$$

Les sous estimateurs suivants peuvent être utilisés:

$$LT(x) + \sum_{i=1}^{bt} b_i w_{B_i} + \sum_{i=1}^{tt} t_i w_{T_i} + \sum_{i=1}^{ft} f_i w_{F_i} + \sum_{i=1}^{ftt} f_{t_i} w_{F_{T_i}} = 0.$$

Où les contraintes d'inégalité appropriées pour les variables w sont ajoutées au problème. Si la contrainte d'égalité non linéaire contient des termes convexes ou des termes généraux non convexes, l'égalité obtenue par la substitution simple des sous estimateurs correspondante est non linéaire.

S'il contient des termes uni variant concave, c'est linéaire mais il correspond à un intervalle réalisable différent. En présence des termes concaves ou des termes unis variant convexes , l'égalité originale $h(x) = 0$ doit donc être réécrite comme deux inégalités de signes opposés.

$$\begin{cases} h(x) \leq 0 \\ -h(x) \leq 0 \end{cases}$$

Ces deux inégalités doivent être sous-estimées indépendamment. Les termes unis variant concaves qui apparaissent dans l'égalité non convexes deviennent convexes en une des deux inégalités pendant que les termes convexes deviennent concaves et les termes non convexes deviennent convexes ou restent non convexes. Le seul obstacle restant à la formulation rigoureuse du problème d'une borne inférieure convexe qui réside dans la sélection des valeurs propres pour les paramètres α dans l'équation (9).

2.3 La construction de la borne inférieure

Le centre de cette section est le développement des méthodes qui produisent une matrice du changement diagonale , c'est à dire un ensemble de paramètres α satisfaisants le Théorème 2.1.Cela autorise la construction d'un sous estimateurs $L(x)$ convexe pour la fonction $f(x)$ deux fois-différentiable sur un domaine indiqué. Deux classes d'approches à ce problème sont définies :

- Le changement uniforme de la diagonale de la matrice Hessienne de $f(x)$.
 - Le changement Hétérogène (non uniforme)de la diagonale de la matrice Hessienne de $f(x)$.
- Notre étude est basée sur la première classe d'approches.

2.3.1 Techniques de calcul de α :

Comme vu dans l'équation (10) et le Théorème 2.1, la matrice du changement diagonale Δ est liée à la matrice Hessienne $Hf(x)$ de la fonction qui est sous-estimée.

Pour les fonctions deux fois-différentiable les éléments de la matrice Hessienne $Hf(x)$ peuvent être des fonctions de variables non linéaires.

Les difficultés qui surviennent de la présence des variables dans la condition de la convexité peuvent être allégées à travers la transformation de la matrice Hessienne dépendante de x exacte à une matrice d'intervalle $[H_f]$ tel que $H_f(x) \subseteq [H_f], \forall x \in [x^L, x^U]$.

On traite les éléments de la matrice Hessienne originale indépendamment en calculant leurs extensions d'intervalle naturels .La famille de matrice Hessienne d'intervalle $[H_f]$ est alors utilisée pour formuler un théorème dans lequel le problème de calcul de α est détendu.

Théorème 2.2.

Soit une fonction générale $f(x)$ avec les dérivés du deuxième ordre continues et sa matrice Hessienne $H_f(x)$.

$L(x)$ est défini par l'équation (9), $[H_f]$ est une matrice d'intervalle symétrique réelle tel que $H_f(x) \subseteq [H_f], \forall x \in [x^L, x^U]$.

Si la matrice $[H_L]$ définie par $[H_L] = [H_f] + 2\Delta = [H_f] + 2diag(\alpha_i)$ est semi-définie positive alors $L(x)$ est convexe sur le domaine $[x^L, x^U]$.

Soit $[x^U, x^L]$ le domaine de validité qui est la condition de la convexité de l'intervalle à travers la matrice d'intervalle.[1],[3],[7]

2.3.2 Évaluation de méthodes de calcul de α :

La qualité du sous-estimateur produite par une méthode du calcul peut être mesurée dans les termes de la distance de la séparation entre la fonction non convexe et son sous-estimateur.

Pour ce but, la distance de la séparation maximale entre $f(x)$ et $L(x)$ est utilisée et montrée qu'elle est directement proportionnelle à α (montré par Maranas et Floudas (1994b)) :

$$d_{max} = \max_{x^L \leq x \leq x^U} (f(x) - L(x)) = \frac{1}{4} \sum_{i=1}^n \alpha_i (x_i^U - x_i^L)^2 \quad (12)$$

En plus, les paramètres et les bornes peuvent être montrés pour affecter le nombre maximum d'itérations exigées pour accomplir l' ϵ -convergence (Maranas et Floudas 1994b) .

La distance de la séparation maximale est utilisée pour évaluer l'exactitude de chaque méthode du calcul.

Exemple 2.1.

Soit à résoudre le problème $\min_{x,y} f(x,y)$ suivant:

$$f(x,y) = \cos x \sin y - \frac{x}{y^2 + 1}$$

$$-1 \leq x \leq 2$$

$$-1 \leq y \leq 1$$

La fonction objectif est la somme de deux termes non linéaires, deux différentes approches sont disponibles pour la dérivation de la matrice Hessienne d'intervalle: dans le cas 1, une simple Hessienne est obtenue pour la fonction objectif alors que dans le cas 2, une matrice Hessienne est tirée pour chaque terme(technique de décomposition).

Cas 1

Basé sur la matrice Hessienne $H_{f(x,y)}$. la matrice Hessienne d'intervalle $[H_f]$ qui contient $H_{f(x,y)}$ est obtenu comme suit:

$$H_{f(x,y)} = \begin{pmatrix} -\cos x \sin y & -\sin x \cos y + \frac{2y}{(y^2+1)^2} \\ -\sin x \cos y + \frac{2y}{(y^2+1)^2} & -\cos x \sin y + \frac{2x(y^2+1)^2 - 8xy^2(y^2+1)}{(y^2+1)^4} \end{pmatrix}$$

Est tel que :

$$H_{f(x,y)} \subseteq [H_f] = \begin{pmatrix} [0.84148, 0.84148] & [-3.00000, 2.84148] \\ [-3.00000, 2.84148] & [-40.84148, 32.84148] \end{pmatrix}$$

Pour $-1 \leq x \leq 2$ et $-1 \leq y \leq 1$.

$[H_f]$ est calculée par les extensions d'intervalle naturels, ce n'est pas la plus petite famille de la hessienne d'intervalle qui contient $H_{f(x,y)}$. Si les valeurs minimales et maximales exactes des éléments de la Hessienne sont calculés par l'optimisation globale (utiliser LINGO) alors les plus petits intervalles exécutables peuvent être identifiés .

Suite à cette procédure, il a été constaté que la matrice Hessienne d'intervalle optimale est en fait:

$$[H_f]^* = \begin{pmatrix} [-0.84148, 0.84148] & [-1.52288, 1.38086] \\ [-1.52288, 1.38086] & [-2.00608, 4.00181] \end{pmatrix}$$

Cas 2

On traite le terme trigonométrique (Terme A) et le terme fractionnel (Terme B) séparément.

Terme A : La matrice Hessienne $H_A(x,y)$ et la matrice Hessienne d'intervalle est $[H_A]$ données par:

$$H_A(x,y) = \begin{pmatrix} -\cos x \sin y & -\sin x \cos y \\ -\sin x \cos y & -\cos x \sin y \end{pmatrix}$$

Et

$$[H_A] = \begin{pmatrix} [-0.84148, 0.84148] & [-1.00000, 0.84148] \\ [-1.00000, 0.84148] & [-0.84148, 0.84148] \end{pmatrix}$$

Terme B :

La matrice Hessienne $H_B(x,y)$ et la matrice Hessienne d'intervalle $[H_B]$ données par:

$$H_B(x,y) = \begin{pmatrix} 0 & \frac{2y}{(y^2+1)^2} \\ \frac{2y}{(y^2+1)^2} & \frac{2x(y^2+1)^2 - 8xy^2(y^2+1)}{(y^2+1)^4} \end{pmatrix}$$

Et

$$[H_B] = \begin{pmatrix} [0,0] & [-2,2] \\ [-2,2] & [-40,32] \end{pmatrix}$$

La matrice Hessienne d'intervalle optimale est :

$$[H_B]^* \begin{pmatrix} [0.00000,0.00000] & [-0.64952,0.64952] \\ [-0.64952,0.64952] & [-2.00000,4.00000] \end{pmatrix}$$

2.3.3 La méthode du changement uniforme de la diagonale de la matrice Hessienne

Pour cette classe de méthodes, le sous-estimateur $L(x)$ est reformulé en utilisant une valeur simple de α :

$$L(x) = f(x) + \alpha \sum_i (x_i^L - x_i)(x_i^U - x_i) \quad (13)$$

Tous les éléments non nuls du changement diagonal de la matrice Δ sont donc égaux à α . Maranas et Floudas (1994b) ont montré que $L(x)$ comme défini par l'équation (13) est convexe si et seulement si:

$$\alpha \geq \max\{0, -\frac{1}{2} \min_{x^L \leq x \leq x^U} \lambda_i(x)\} \quad (14)$$

Où $\lambda_i(x)$ sont les valeurs propres de la matrice Hessienne $H_f(x)$.

Si $f(x)$ est convexe, toutes les valeurs propres de $H_f(x)$ sont positifs pour n'importe quel $x \in [x^L, x^U]$ et par l'équation (14) $\alpha = 0$: la fonction originale semble inchangée dans le problème de borne inférieure.

Sinon α augmente comme les valeurs propres négatives de même $f(x)$ devient de plus en plus non convexas. Le problème de minimisation qui apparaît dans l'équation (14) peut être écrit explicitement comme :

$$\begin{cases} \min_{x,\lambda} \lambda, \\ H_f(x) - \lambda I = 0, \\ x \in [x^L, x^U], \end{cases}$$

Où I la matrice d'identité.

Maranas et Floudas (1994) ont suggérés que la plus petite valeur propre de $H_f(x)$ pourrait être obtenue en utilisant la matrice Hessienne pour un problème d'optimisation.

Pourtant, cette approche exige la solution d'un problème de programmation convexe basé sur les dérivés du deuxième ordre de la fonction pour être sous-estimée et implique une grande quantité d'effort.

La plus petite valeur propre de $H_f(x)$ peut être facilement obtenue quand la matrice Hessienne d'intervalle $[H_f] \supseteq H_f(x)$ est présentée. La méthode présentées dans cette section produit une valeur simple de α qui satisfait la condition suffisante suivante pour la convexité de $L(x)$:

$$\alpha \geq \{0, -\frac{1}{2}\lambda_{\min}([H_f])\} \quad (16)$$

Où λ_{\min} est la valeur propre minimale de la matrice d'intervalle $[H_f]$.

Les deux méthodes $\theta(n^2)$ et $\theta(n^3)$ s'utilisent pour calculer la valeur propre minimale d'une matrice d'intervalle symétrique .

Dans notre étude on s'intéresse qu'aux deux méthodes de $\theta(n^3)$.

2.3.4 Méthode basée sur le théorème de Kharitonov

Cette méthode (kharitonov, 1979)est utilisée pour déterminer si un polynôme d'intervalle $P(\lambda)$ est stable ou pas en testant seulement la stabilité de quatre polynômes.

Adjiman et al (1996) on montrer que le théorème de kharitonov ne peut pas être utilisé seulement pour déterminer la stabilité des polynômes mais aussi pour calculer la valeur de λ_{\min} .

La technique proposée pour le calcul de la valeur propre minimale λ_{\min} d'une fonction arbitraire deux-fois-différentiable $f(x)$ est fondé sur la dérivation de son polynôme caractéristique d'intervalle.

Théorème 2.3.

un intervalle polynômial $P(\lambda)$ définie par :

$$P(\lambda) = [a_0^L, a_0^U] + [a_1^L, a_1^U]\lambda + \dots + [a_{n-1}^L, a_{n-1}^U]\lambda^{n-1} + \lambda^n$$

Où $a_i^L \leq a_i^U, \forall i$.

$P_4(\lambda)$ dénote le sous-ensemble qui contient les quatre polynômes (polynômes Kharitonov):

$$K_1(f, X, \lambda) = a_0^L + a_1^L \lambda + a_2^U \lambda^2 + a_3^U \lambda^3 + a_4^L \lambda^4 + a_5^L \lambda^5 + a_6^U \lambda^6 + \dots$$

$$K_2(f, X, \lambda) = a_0^U + a_1^U \lambda + a_2^L \lambda^2 + a_3^L \lambda^3 + a_4^U \lambda^4 + a_5^U \lambda^5 + a_6^L \lambda^6 + \dots$$

$$K_3(f, X, \lambda) = a_0^U + a_1^U \lambda + a_2^L \lambda^2 + a_3^U \lambda^3 + a_4^U \lambda^4 + a_5^L \lambda^5 + a_6^L \lambda^6 + \dots$$

$$K_4(f, X, \lambda) = a_0^L + a_1^U \lambda + a_2^U \lambda^2 + a_3^L \lambda^3 + a_4^L \lambda^4 + a_5^U \lambda^5 + a_6^U \lambda^6 + \dots$$

Alors $P(\lambda)$ et $P_4(\lambda)$ ont la même λ_{min} .

Ce résultat diminue grandement la complexité de λ_{min} car le nombre de polynômes considéré est réduit à quatre.

La procédure suivante peut être utilisée alors pour calculer une borne inférieure d'une valeur propre de la matrice Hessienne $H(x)$:

1. Construisez $H(X) - \lambda I$ où I est l'identité de la matrice.
2. Dérivez le déterminant de $H(X) - \lambda I$ et la mise à zéro. Le polynôme résultant est de la forme:

$$P(X, \lambda) = a_0(X) + a_1(X)\lambda + a_2(X)\lambda^2 + a_3(X)\lambda^3 + \dots$$

3. Utilisant l'arithmétique de l'intervalle, on obtient un intervalle $P(\lambda)$ de polynomial pour lequel il contient $P(X, \lambda)$.
4. Utilisant Théorème 2.3, calculez λ_{min} (la borne inférieure qui est la valeur propre minimale de $H(x)$).

Le recours à l'arithmétique d'intervalle dans le Pas 3 est nécessaire car il transforme énormément de difficulté.

Cependant, les familles de polynômes sont agrandis par le processus et comme λ_{min} est la racine du polynôme $P(\lambda)$ il ne peut pas être la racine des polynômes de $P(X, \lambda)$. Donc, la valeur obtenue est une borne inférieure qui est la valeur propre minimale.

Une alternative de la procédure présentée au-dessus est exécutée on utilisant l'extensions de l'intervalle sur la matrice Hessienne , avant le calcul de déterminant il est possible que cela aggrave le problème de l'exactitude. Si on commence de la matrice Hessienne exacte ou d'une matrice Hessienne qui le contient, le polynôme d'intervalle peut produire un plus grand spectre de valeur propre que l'intervalle de la matrice Hessienne. voir [3],[7].

Exemple 2.2.

On considère $f(x_1, x_2) = x_1^3 - x_1 x_2^2$ avec $(x_1, x_2) \in X = [0, 1]^2$.

Son polynôme caractéristique est:

$$-12x_1^2 - 4x_2^2 - 4x_1\lambda + \lambda^2 = 0$$

Son polynôme caractéristique d'intervalle sur X est:

$$[-16,0] + [-4,0]\lambda + \lambda^2 = 0.$$

Les quatre polynômes de Kharitonov sont alors :

$$K_1(f,X,\lambda) = -16 - 4\lambda + \lambda^2$$

$$K_2(f,X,\lambda) = \lambda^2$$

$$K_3(f,X,\lambda) = -4 + \lambda^2$$

$$K_4(f,X,\lambda) = -16 + \lambda^2$$

L'ensemble des racines de ces polynômes est $-4, 2 - 2\sqrt{5}, 4, 2 + 2\sqrt{5}$ Ainsi le calcul de la plus petite valeur propre minimale de $f(x_1, x_2)$ sur X est -4 .

2.4 La Méthode de la matrice Hessienne d'intervalle (Hertz's method)

Cette méthode est décrite en détail par Hertz (1992) , Adjiman et Floudas (1996). C'est semblable au théorème de Kharitonov: un sous-ensemble définit d'une matrices est construit de la matrice d'intervalle, avec la propriété que la valeur propre minimale est égal à la valeur propre minimale de la matrice d'intervalle. Les matrices exigées peuvent être obtenu à travers une procédure systématique.

Théorème 2.4.

$H^* = ([a_{ij}^L, a_{ij}^U])$ est un intervalle symétrique de la matrice Hessienne.

Soit $X \in R^n$ un vecteur. Il y a 2^{n-1} combinaisons possibles pour les signes du $x_i x_j$ produits ($i \notin j$).

Soit la matrice du sommet $H_K^* \in H^*$ définie par:

$$H_K^* = a_{ij}^K ; \forall K \in \{1, \dots, 2^{n-1}\}. \quad (17)$$

Où

$$a_{ij}^K = \begin{cases} a_{ij}^L & \text{si } i = j \\ a_{ij}^L & \text{si } x_i x_j \geq 0, i \neq j \\ a_{ij}^U & \text{si } x_i x_j \leq 0, i \neq j \end{cases} \quad (18)$$

Alors la plus petite valeur propre de la matrices H_K^* est une valeur propre de H^* .

Preuve.

Il a été montré que les valeurs propres de de matrice $H_{f(x)}$ est bornée par les valeurs propres extrêmes d'un sous-ensemble de ces matrices.

Pour déterminer une borne inférieure de la plus petite valeur propre λ_{min} , la valeur propre minimale de la matrices c'est le calcul de 2^{n-1} sommet de $H_{f(x)}$.

Par définition, λ_{min} peut être exprimé comme :

$$\lambda_{min} = \min_{H \in H_{f(x)}} (\min_{\|r\|=1} r^T H r) \quad (19)$$

Où r est un vecteur réel de dimension n et H est une matrice des éléments réels , a_{11}, \dots, a_{nn} appartenant à la matrice d'intervalle $H_{f(x)}$.

Noté que l'ensemble de tous les vecteurs r couvre la surface de la sphère unitaire. Les défis présentés par le problème (19) peuvent être survenus en imaginant des règles simples qui identifient a priori la matrice réelle $H^* \in H_{f(x)}$ qui minimisent la forme quadratique $r^T H r$ pour tous les vecteurs r satisfaisant $\|r\| = 1$.

L'équation (19) peut être réécrite comme :

$$\lambda_{min} = \min_{H \in H_{f(x)}} (\min_{\|r\|=1} \sum_{i=1}^n a_{ii} r_i^2 + \sum_{1 \leq j, i \leq n, i \neq j} a_{ij} r_i r_j) \quad (20)$$

$$= \min_{H \in H_{f(x)}} (\min_{\|r\|=1} \sum_{i=1}^n a_{ii} r_i^2 + \min_{\|r\|=1} \sum_{1 \leq j, i \leq n, i \neq j} a_{ij} r_i r_j) \quad (21)$$

$$= \min_{\|r\|=1} (\sum_{i=1}^n a_{ii}^L r_i^2) + \min_{H \in H_{f(x)}} (\min_{\|r\|=1} \sum_{1 \leq j, i \leq n, i \neq j} a_{ij} r_i r_j) \quad (22)$$

Donc le premier terme a été réduit à une minimisation de la surface de la sphère unitaire et les éléments de la diagonale de la matrice H^* désirée ont été identifiés comme les bornes

inférieures des éléments diagonaux de la matrice d'intervalle de $H_{f(x)}$.

Le deuxième terme pourrait être simplifié dans une manière semblable si le signe du $r_i r_j$ du produit était su.

Donc si $r_i r_j < 0$, le (ij) et (ji) les éléments de H^* devraient être égaux à a_{ij}^U .
 si $r_i r_j \geq 0$ ils devraient être égaux à a_{ij}^L .

Cette réduction peut être accomplie en notant que l'espace du n-dimension peut être divisé en 2^n sommets dans lesquels les signes des éléments de vecteur r restent inchangés. Dans chacun de ces sommets , le signe de chaque produit $r_i r_j, (i, j) \in \{1, \dots, \}^2$ est aussi conservé.

De plus, dû à la symétrie au sujet de l'origine du coordonné du système, il y a seulement 2^{n-1} combinaisons possibles des signes de ce produit en corollaire afin que 2^{n-1} sommets soient suffisants pour décrire tout le signe qui conserve des domaines des produits $r_i r_j$.

R_K est un ensemble de vecteurs r .

Alors pour chaque sommet k, la matrice réelle $H_K^* \in H_{f(x)}$ minimise $r^T H r$ peut être construite d'après la règle suivante:

$$H_K^* = a_{ij}^K \quad ; \forall K \in \{1, \dots, 2^{n-1}\}.$$

Où

$$a_{ij}^K = \begin{cases} a_{ij}^L & \text{si } i = j \\ a_{ij}^L & \text{si } x_i x_j \geq 0, i \neq j \\ a_{ij}^U & \text{si } x_i x_j \leq 0, i \neq j \end{cases}$$

Par conséquent, la valeur propre minimale de la famille de matrices $H_{f,x}$ est donné par :

$$\begin{aligned} \lambda_{min} &= \min_{K=1, \dots, 2^{n-1}} (\min_{r \in R_K} r^T H_K^* r) \quad (23) \\ &= \min_{K=1, \dots, 2^{n-1}} \lambda_{K, min} \end{aligned}$$

Où $\lambda_{K, min}$ est la valeur propre minimale de H_K^* de la matrice comme défini dans l'équation (18).

Noté que tous les 2^{n-1} de H^* sont des sommets de la matrices qui ont leurs éléments aux extrémités des intervalles qui apparaissent dans la matrice $H_{f,X}$.

En outre, ces matrices sont symétriques ($r_i r_j = r_j r_i$) et leurs valeurs propres sont par conséquent réelles.

Exemple 2.3.

Le polynôme de troisième ordre dans deux variables est étudié comme suit :

$$f(x_1, x_2) = x_1^3 - x_1 x_2^2 \text{ avec } (x_1, x_2) \in X = [0, 1]^2 .$$

Sa matrice Hessienne est :

$$H_{f(x_1, x_2)} = \begin{pmatrix} 6x_1 & -2x_2 \\ -2x_2 & -2x_1 \end{pmatrix} \subseteq H_{f,x} = \begin{pmatrix} [0, 6] & [-2, 0] \\ [-2, 0] & [-2, 0] \end{pmatrix} .$$

Les deux matrices du sommet sont alors:

$$H_1^* = \begin{pmatrix} 0 & -2 \\ -2 & -2 \end{pmatrix} .$$

Et

$$H_2^* = \begin{pmatrix} 0 & 0 \\ 0 & -2 \end{pmatrix} .$$

Et leurs plus petites valeurs propres sont $\lambda_{1,min} = -1 - \sqrt{5}$ et $\lambda_{2,min} = -2$ d'où $\lambda_{min} \geq -3.24$.

Exemple 2.4.

Le deuxième exemple présenté ici est une fonction de deux variables qui implique des termes trigonométriques:

$$f(x_1, x_2) = x_1 \cos x_2 + x_2 \sin x_1 \text{ avec } (x_1, x_2) \in X = [0, 1]^2 .$$

La matrice Hessienne correspondante est :

$$H_{(f(x_1, x_2))} = \begin{pmatrix} -x_1 \sin x_2 & -\sin x_2 + \cos x_1 \\ -\sin x_2 + \cos x_1 & -x_1 \cos x_2 \end{pmatrix} \subseteq H_{f,x} = \begin{pmatrix} [-0.842, 0] & [-0.302, 1] \\ [-0.302, 1] & [-1, 0] \end{pmatrix}$$

Deux matrices du sommet sont exigées pour évaluer la plus petite valeur propre

$$H_1^* = \begin{pmatrix} -0.842 & -0.302 \\ -0.302 & -1 \end{pmatrix}.$$

Et

$$H_2^* = \begin{pmatrix} -0.842 & 1 \\ 1 & -1 \end{pmatrix}.$$

Les deux valeurs propres minimales sont:

$$\lambda_{1,min} = -1.234 \text{ et } \lambda_{2,min} = -1.925.$$

Cela cède une borne inférieure de -1.925 sur la plus petite valeur propre de $H_{f(x_1,x_2)}$ sur X.

2.5 Relaxation de la fonction borne inférieure

Considérons l'ensemble $H \subset R^n$ et un point $x^j \in H$, où la fonction deux fois différentiable f est définie.

La fonction borne inférieure pour f est :

$$L(x^j) = f(x^j) + \alpha \sum_i (x_i^L - x_i^j)(x_i^U - x_i^j)$$

La borne inférieure pour f est le minimum de $f^i(x)$ sur H.

Si H est un hyper rectangle I tel que :

$$I = [a,b]$$

et

$$x^j = \frac{a+b}{2} = c.$$

c est le centre de I, cette borne inférieure est définie par:

$$\bar{L} = \min_{x \in I} f^j(x) = f(c) + \alpha \sum_i (x_i^L - c)(x_i^U - c)$$

Supposons K points $(x^j)_{j=1,2,\dots,k}$ sont donnés dans I , la fonction borne inférieure pour la fonction f est:

$$F^k(x) = \min_{j=1,2,\dots,k} f^j(x)$$

Considérons la partition de I en K hyper rectangle $(I_j)_{j=1,2,\dots,k}$ et on note par c^j le centre de chaque hyper rectangle I_j .

$$\forall j = 1,2,\dots,k, \forall x \in I_j, F_2^k(x) = f(c^j) + \alpha \sum_i (x_i^L - c_i^j)(x_i^U - c_i^j)$$

La fonction borne inférieure est constante sur tout les hyper rectangle

2.6 Algorithme α BB

1. Soit $X \in H$, avec $X = (x_1, x_2, \dots, x_n)$ et $H = [a^j, b^j] = [a^{1^j}, b^{1^j}] \times [a^{2^j}, b^{2^j}] \times [a^{3^j}, b^{3^j}] \times \dots \times [a^{n^j}, b^{n^j}] \subseteq R^n$.

avec $j = 0, 1, 2, \dots, k; k \in N$.

2. Posons $j := k, I_K = H$, fixer $\epsilon > 0, \alpha > 0$.

3. Calcul de $c^{1^k}, c^{2^k}, c^{3^k}, \dots, c^{n^k}, X = (c^{1^k}, c^{2^k}, c^{3^k}, \dots, c^{n^k}), LB_k$ et UB_k .

$$\begin{cases} x_1 \in [a^{1^k}, b^{1^k}] \\ x_2 \in [a^{2^k}, b^{2^k}] \\ x_3 \in [a^{3^k}, b^{3^k}] \\ \vdots \\ x_n \in [a^{n^k}, b^{n^k}] \end{cases} \Rightarrow \begin{cases} c^{1^k} = \frac{a^{1^k} + b^{1^k}}{2} \\ c^{2^k} = \frac{a^{2^k} + b^{2^k}}{2} \\ c^{3^k} = \frac{a^{3^k} + b^{3^k}}{2} \\ \vdots \\ c^{n^k} = \frac{a^{n^k} + b^{n^k}}{2} \end{cases} \Rightarrow \begin{cases} X^k = (c^{1^k}, c^{2^k}, c^{3^k}, \dots, c^{n^k}) \\ UB_k = f(X^k) \\ LB_k = f(X^k) + \alpha \sum_i (x_i^L - x_i^k)(x_i^U - x_i^k) \end{cases}$$

4. Si: $UB_K - LB_K \leq \epsilon$, on s'arrête et on aura la solution du problème (p):

$$\begin{cases} X^k = \hat{X} \\ UB_k = \min f(X) \end{cases}$$

5. Sinon: subdiviser I_K en deux sous intervalles

(ou en un nombre finie de sous intervalles) I_{K_1} et I_{K_2} tel que:

$$\bigsqcup_{i=1}^{i=2} I_{K_i} = I \text{ et } \dot{I}_{K_1} \cap \dot{I}_{K_2} = \emptyset.$$

Où \dot{I} est l'intérieur de R.

6.construire les problèmes des bornes inférieures et supérieures de $\min f(x)$ sur $H \cap I_{K_i}$; $i=1,2$.

Soient : LB_{K_1}, UB_{K_1} et LB_{K_2}, UB_{K_2} les solutions obtenues .

7. Posons :

$$\begin{cases} UB_{K+1} = \min\{UB_{K_1}, UB_{K_2}, UB_K\}, \\ LB_{K+1} = \min\{LB_{K_1}, LB_{K_2}\} = LB_{K^*}, \end{cases}$$

8. Posons: $I_K = \{R_{K_1}, R_{K_2}\}$.

9. Éliminer de I_K tous sous ensembles $I_{K_j}, j=1,2$ tel que $LB_{K_j} > UB_{K+1}$ où $H \cap I_{K_j} = \emptyset$. et posons : $I_{K+1} = I_{K_{i^*}}$.

10. Posons : $K = k + 1$ et revenant à 4-10.

Notation :

Soit : $I_k = \{x^{j^k} \in R : a^{j^k} \preceq x^{j^k} \preceq b^{j^k}, (j = 1, 2, \dots, k), (k = 1, 2, \dots)\}$: l'intervalle à l'itération k .

$F^k(x) = \max\{f^j(x), x \in I_k\}$: l'enveloppe convexe supérieure .

X^k : la solution optimale globale du problème (P^k) correspondant $(I_k = R, F^k)$, alors R sera subdivisé en deux sous intervalles R_{k_1} et R_{k_2} tels que :

$$R_{k_1} = \{x^{j^k} \in R : a^{j^k} \preceq x^{j^k} \preceq b^{j^k}, (j = 1, 2, \dots, k), (k = 1, 2, \dots)\}$$

$$R_{k_2} = \{x^{j^k} \in R : a^{j^k} \preceq x^{j^k} \preceq b^{j^k}, (j = 1, 2, \dots, k), (k = 1, 2, \dots)\}$$

2.6.1 La convergence de l'algorithme α BB

Il existe plusieurs propositions qui montre la convergence de l'algorithme α BB , parmi ces propositions:

Preuve. (Gourdin et al)

Soit l'algorithme s'arrête ,soit il converge asymptotiquement

$$\lim_{k \rightarrow -\infty} f_{opt}^k = \lim_{k \rightarrow -\infty} F_{opt}^k = f^* = \min_{x \in I} f(x).$$

L'algorithme s'arrête après un nombre fini d'itérations quand $\epsilon=0$.

Une telle situation se produit quand , à une étape donnée, la fonction objectif coincide avec la fonction borne inférieure au moins dans un voisinage du minimum global.

$$f^* = \min_{x \in I} f(x)$$

autrement dit ,l'algorithme est asymptotiquement convergent.

2.7 Conclusion

Comme c'est démontrer dans ce chapitre l'algorithme α BB peut identifier rigoureusement la solution globale de problème de la programmation non convexe deux fois différentiable basé sur le plan de la sous estimation convexe .

Deux méthodes ont été développées pour le calcul du paramètre α nécessaire pour la construction d'un sous-estimateur convexe . En utilisant l'approche du changement de la diagonale pour lequel il exige le calcul d'une borne inférieure sur la valeur propre d'une matrice hessienne d'intervalle .

La décomposition des fonctions non linéaires en une somme de termes constitue un concept central pour la généralisation d'un sous estimateur .En général les termes non convexes devraient impliquer peu de variables on réduisant les coûts de calcul de α .

De plus la construction des fonctions de borne inférieures personnalisées pour différentes classes des termes tels que les termes bi linéaires ,tri linéaires ,fractionnaires,tri linéaires fractionnaires ou uni variants concaves peut rehausser les caractéristiques de la convergence de α BB.

Chapitre 3

Exemples d'applications

3.0.1 Introduction

Dans ce chapitre nous avons traités deux exemples de fonctions définies sur un hyper rectangle de R^2 , avec un paramètre α calculé (voir chapitre 2), on utilisant la méthode Branch-and-Bound dans le but de monter l'efficacité de cet algorithme .

(Le premier et le deuxième exemple sont tirés de l'article [1]).

3.0.2 Application de la méthode Branch-and-Bound à l'optimisation globale des fonctions deux fois différentiable dans R^n

Algorithme α BB

1. Soit $X \in H$, avec $X = (x_1, x_2, \dots, x_n)$ et
 $H = [a^j, b^j] = [a^{1^j}, b^{1^j}] \times [a^{2^j}, b^{2^j}] \times [a^{3^j}, b^{3^j}] \times \dots \times [a^{n^j}, b^{n^j}] \subseteq R^n$.
avec $j = 0, 1, 2, \dots, k; k \in N$.

2. Posons $j := k, I_K = H$, fixer $\epsilon > 0, \alpha > 0$.

3. Calcul de : $c^{1^k}, c^{2^k}, c^{3^k}, \dots, c^{n^k}, X = (c^{1^k}, c^{2^k}, c^{3^k}, \dots, c^{n^k}), LB_k$ et UB_k .

$$\left\{ \begin{array}{l} x_1 \in [a^{1^k}, b^{1^k}] \\ x_2 \in [a^{2^k}, b^{2^k}] \\ x_3 \in [a^{3^k}, b^{3^k}] \\ \vdots \\ x_n \in [a^{n^k}, b^{n^k}] \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} c^{1^k} = \frac{a^{1^k} + b^{1^k}}{2} \\ c^{2^k} = \frac{a^{2^k} + b^{2^k}}{2} \\ c^{3^k} = \frac{a^{3^k} + b^{3^k}}{2} \\ \vdots \\ c^{n^k} = \frac{a^{n^k} + b^{n^k}}{2} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} X^k = (c^{1^k}, c^{2^k}, c^{3^k}, \dots, c^{n^k}) \\ UB_k = f(X^k) \\ LB_k = f(X^k) + \alpha \sum_i (x_i^L - x_i^k)(x_i^U - x_i^k) \end{array} \right.$$

4. Si: $UB_K - LB_K \leq \epsilon$, on s'arrête et on aura la solution du problème (p):

$$\begin{cases} X^k = \hat{X} \\ UB_k = \min f(X) \end{cases}$$

5. Sinon: subdiviser I_K en deux sous intervalles (ou en un nombre finie de sous intervalles)

I_{K_1} et I_{K_2} tel que:

$$\bigsqcup_{i=1}^{i=2} I_{K_i} = I \text{ et } \dot{I}_{K_1} \cap \dot{I}_{K_2} = \emptyset.$$

Où \dot{I} est l'intérieur de R .

6. construire les problèmes des bornes inférieures et supérieures de $\min f(x)$ sur $H \cap I_{K_i}$, $i=1,2$.

Soient : LB_{K_1}, UB_{K_1} et LB_{K_2}, UB_{K_2} les solutions obtenues .

7. Posons :

$$\begin{cases} UB_{K+1} = \min\{UB_{K_1}, UB_{K_2}, UB_K\} \\ LB_{K+1} = \min\{LB_{K_1}, LB_{K_2}\} = LB_{K^*} \end{cases}$$

8. Posons: $I_K = \{R_{K_1}, R_{K_2}\}$.

9. Éliminer de I_K tous sous ensembles $I_{K_j}; j=1,2$ tel que:

$$LB_{K_j} > UB_{K+1} \text{ où } H \cap I_{K_j} = \emptyset.$$

et posons : $I_{K+1} = I_{K_i^*}$.

10. Posons : $K = k + 1$ et revenant à 4-10.

Application numérique de l'algorithme

Exemple 3.1.

Soit à résoudre le problème suivant:

$$(P) \begin{cases} \min f(x) \\ X \in H \end{cases}$$

$$\text{Où } f(x_1, x_2) = x_1^3 - x_1 x_2^2$$

$$X = (x_1, x_2) \in H, H = [0,1] \times [0,1], \alpha = 1.25, \epsilon = 10^{-5}$$

0 Itération:(j=0)

$$1. \text{ Soit } (x_1, x_2) \in [a^j, b^j] \times [a^{j'}, b^{j'}]$$

$$2. \text{ Posons : } j := 0, I_0 = [a^0, b^0] \times [a^{0'}, b^{0'}] = [0,1] \times [0,1]$$

$$3. \text{ Calcul de } c^0, c^{0'}, x_0, UB_0 \text{ et } LB_0 :$$

$$\begin{cases} x_1 \in [0,1] \\ x_2 \in [0,1] \end{cases} \Rightarrow \begin{cases} c^0 = \frac{a^0+b^0}{2} = \frac{1}{2} \\ c^{0'} = \frac{a^{0'}+b^{0'}}{2} = \frac{1}{2} \end{cases} \Rightarrow \begin{cases} X_0 = (0.5,0.5) \\ UB_0 = 0 \\ LB_0 = -0.81 \end{cases}$$

4. Test d'arrêt :

$UB_0 - LB_0 = 0.81 > \epsilon$, donc $\min f(X) \neq UB_0$ et $\hat{X} \neq x_0$, on passe à la première itération, c.à.d:

subdiviser l'hyperrectangle $I_0 = [0,1] \times [0,1]$ en deux sous hyper rectangle suivant le plus grand coté

$$\begin{cases} x_1 \in [0,1] \\ x_2 \in [0,1] \end{cases} \Rightarrow \begin{cases} b^0 - a^0 = 1 - 0 = 1 \\ b^{0'} - a^{0'} = 1 - 0 = 1 \end{cases}$$

\Rightarrow comme les deux cotés sont égaux, donc on choisie x, puis on subdivise l'hyper rectangle $[0,1] \times [0,1]$ en deux hyper rectangles $I_{1_1} = [0, \frac{1}{2}] \times [0,1]$, $I_{1_2} = [\frac{1}{2}, 1] \times [0,1]$

1 itération : (j=1)

1. Soit $(x_1, x_2) \in [a^j, b^j] \times [a^{j'}, b^{j'}]$

2. Posons : $j := 1$, $I_1 = [a^1, b^1] \times [a^{1'}, b^{1'}] = [0,1] \times [0,1]$, $\epsilon = 10^{-5}$

1^{ier} Cas: $x \in [a_{1_1}, b_{1_1}] \times [a'_{1_1}, b'_{1_1}] = [0, \frac{1}{2}] \times [0,1]$

(a) Calcul de c_{1_1} , c'_{1_1} , X_{1_1} , UB_{1_1} et LB_{1_1} :

$$\begin{cases} x_1 \in [0, \frac{1}{2}] \\ x_2 \in [0,1] \end{cases} \Rightarrow \begin{cases} c_{1_1} = \frac{a_{1_1}+b_{1_1}}{2} = \frac{1}{4} \\ c'_{1_1} = \frac{a'_{1_1}+b'_{1_1}}{2} = \frac{1}{2} \end{cases} \Rightarrow \begin{cases} X_{1_1} = (0.25, 0.5) \\ UB_{1_1} = \frac{-3}{64} = -0.047 \\ LB_{1_1} = -0.553 \end{cases}$$

2^{ieme} Cas: $x \in [a_{1_2}, b_{1_2}] \times [a'_{1_2}, b'_{1_2}] = [\frac{1}{2}, 1] \times [0,1]$

(a) Calcul de c_{1_2} , c'_{1_2} , X_{1_2} , UB_{1_2} et LB_{1_2} :

$$\begin{cases} x_1 \in [\frac{1}{2}, 1] \\ x_2 \in [0,1] \end{cases} \Rightarrow \begin{cases} c_{1_2} = \frac{a_{1_2}+b_{1_2}}{2} = \frac{3}{4} = 0.75 \\ c'_{1_2} = \frac{a'_{1_2}+b'_{1_2}}{2} = \frac{1}{2} = 0.5 \end{cases} \Rightarrow \begin{cases} X_{1_2} = (0.75, 0.5) \\ UB_{1_2} = 0.233 \\ LB_{1_2} = -0.071 \end{cases}$$

3. Calcul de LB_1 et UB_1 :

$$\begin{cases} UB_1 = \min\{UB_{1_1}, UB_{1_2}, UB_0\} = -0.047 \\ LB_1 = \min\{LB_{1_1}, LB_{1_2}\} = -0.553 \end{cases}$$

4. Test d'arrêt:

$UB_1 - LB_1 = 0.5 > \epsilon$, donc $\min f(X) \neq UB_1$ et $\hat{X} \neq X_{1_1}$ et $\hat{X} \neq X_{1_2}$, on passe à la

deuxième itération , c.à.d:

subdiviser l'hyperrectangle $I_{11} = [0, \frac{1}{2}] \times [0, 1]$ en deux sous hyper rectangle suivant le plus grand coté car $LB_1 = LB_{11}$

$$\begin{cases} x_1 \in [0, \frac{1}{2}] \\ x_2 \in [0, 1] \end{cases} \Rightarrow \begin{cases} b^{11} - a^{11} = \frac{1}{2} - 0 = \frac{1}{2} \\ b^{1'1} - a^{1'1} = 1 - 0 = 1 \end{cases}$$

\Rightarrow comme les deux cotés ne sont pas égaux ,alors on subdivise suivant le plus grand coté qui est y l'hyperrectangle $[0, \frac{1}{2}] \times [0, 1]$ en deux hyper rectangles $I_{21} = [0, \frac{1}{2}] \times [\frac{1}{2}, 1]$, $I_{22} = [0, \frac{1}{2}] \times [0, \frac{1}{2}]$, et on laisse l'hyper rectangle $[0, 1] \times [0, 1]$.

5.Pas d'élimination car les $LB_{k_j} < UB_{k+1}$.

2^{ieme} Itération :(j=2)

1.Soit $(x_1, x_2) \in [a^j, b^j] \times [a^{j'}, b^{j'}]$

2.Posons : $j := 2, I_2 = [a^2, b^2] \times [a^{2'}, b^{2'}] = [\frac{1}{2}, 1] \times [0, 1], \epsilon = 10^{-5}$

1^{ier} Cas: $x \in [a_{21}, b_{21}] \times [a'_{21}, b'_{21}] = [0, \frac{1}{2}] \times [\frac{1}{2}, 1]$

(a) Calcul de $c_{21}, c'_{21}, X_{21}, UB_{21}$ et LB_{21} :

$$\begin{cases} x_1 \in [0, \frac{1}{2}] \\ x_2 \in [\frac{1}{2}, 1] \end{cases} \Rightarrow \begin{cases} c_{21} = \frac{a_{21} + b_{21}}{2} = \frac{1}{4} \\ c'_{21} = \frac{a'_{21} + b'_{21}}{2} = \frac{3}{4} \end{cases} \Rightarrow \begin{cases} X_{21} = (0.25, 0.75) \\ UB_{21} = -0.125 \\ LB_{21} = -0.125 \end{cases}$$

2^{ieme} Cas: $x \in [a_{22}, b_{22}] \times [a'_{22}, b'_{22}] = [0, \frac{1}{2}] \times [0, \frac{1}{2}]$

(a) Calcul de $c_{22}, c'_{22}, X_{22}, UB_{22}$ et LB_{22} :

$$\begin{cases} x_1 \in [0, \frac{1}{2}] \\ x_2 \in [0, \frac{1}{2}] \end{cases} \Rightarrow \begin{cases} c_{22} = \frac{a_{22} + b_{22}}{2} = \frac{1}{4} = 0.25 \\ c'_{22} = \frac{a'_{22} + b'_{22}}{2} = \frac{1}{4} = 0.25 \end{cases} \Rightarrow \begin{cases} X_{22} = (0.25, 0.25) \\ UB_{22} = 0 \\ LB_{22} = 0 \end{cases}$$

3.Calcul de LB_2 et UB_2 :

$$\begin{cases} UB_2 = \min\{UB_{21}, UB_{22}, UB_1\} = -0.125 \\ LB_2 = \min\{LB_{21}, LB_{22}\} = -0.125 \end{cases}$$

4. Test d'arrêt:

$UB_2 - LB_2 = 0 < \epsilon$, donc $UB_k = \min f(X)$ et $X^k = \hat{X}$.

Tableau représentatif des résultats du premier exemple.

| K | x^{K_1} | x^{K_2} | UB_{K_1} | UB_{K_2} | UB_K | LB_{K_1} | UB_{K_2} | UB_K |
|-----|-------------|-------------|------------|------------|--------|------------|------------|--------|
| 0 | (0.5,0.5) | | 0 | | 0 | -0.81 | | -0.81 |
| 1 | (0.25,0.5) | (0.75,0.5) | -0.047 | 0.233 | -0.047 | -0.553 | -0.071 | -0.553 |
| 2 | (0.25,0.75) | (0.25,0.25) | -0.125 | 0 | -0.125 | -0.125 | 0 | -0.125 |

Exemple 3.2.

$f(x_1, x_2) = x_1 \cos x_2 + x_2 \sin x_1$ avec $(x_1, x_2) \in H, H = [0,1] \times [0,1]$

$\alpha = 0.635, \epsilon = 10^{-5}$

0 Itération:(j=0)

1. Soit $(x_1, x_2) \in [a^j, b^j] \times [a^{j'}, b^{j'}]$

2. Posons : $j := 0, I_0 = [a^0, b^0] \times [a^{0'}, b^{0'}] = [0,1] \times [0,1]$

3. Calcul de $c^0, c^{0'}, x_0, UB_0$ et LB_0 :

$$\begin{cases} x_1 \in [0,1] \\ x_2 \in [0,1] \end{cases} \Rightarrow \begin{cases} c^0 = \frac{a^0 + b^0}{2} = \frac{1}{2} \\ c^{0'} = \frac{a^{0'} + b^{0'}}{2} = \frac{1}{2} \end{cases} \Rightarrow \begin{cases} X_0 = (0.5, 0.5) \\ UB_0 = 0.538 \\ LB_0 = 0.220 \end{cases}$$

4. Test d'arrêt :

$UB_0 - LB_0 = 0.318 > \epsilon$, donc $\min f(X) \neq UB_0$ et $\hat{X} \neq x_0$, on passe à la première itération, c.à.d:

subdiviser l'hyperrectangle $I_0 = [0,1] \times [0,1]$ en deux sous hyper rectangle suivant le plus grand coté

$$\begin{cases} x_1 \in [0,1] \\ x_2 \in [0,1] \end{cases} \Rightarrow \begin{cases} b^0 - a^0 = 1 - 0 = 1 \\ b^{0'} - a^{0'} = 1 - 0 = 1 \end{cases}$$

\Rightarrow comme les deux cotés sont égaux, donc on choisie x , puis on subdivise l'hyper rectangle $[0,1] \times [0,1]$ en deux hyper rectangles $I_{1_1} = [0, \frac{1}{2}] \times [0,1], I_{1_2} = [\frac{1}{2}, 1] \times [0,1]$

1 itération:(j=1)

1. Soit $(x_1, x_2) \in [a^j, b^j] \times [a^{j'}, b^{j'}]$

2. Posons : $j := 1, I_1 = [a^1, b^1] \times [a^{1'}, b^{1'}] = [0,1] \times [0,1], \epsilon = 10^{-5}$

^{1^{ier}} Cas: $x \in [a_{1_1}, b_{1_1}] \times [a'_{1_1}, b'_{1_1}] = [0, \frac{1}{2}] \times [0,1]$

(a) Calcul de c_{1_1} , c'_{1_1} , X_{1_1} , UB_{1_1} et LB_{1_1} :

$$\begin{cases} x_1 \in [0, \frac{1}{2}] \\ x_2 \in [0, 1] \end{cases} \Rightarrow \begin{cases} c_{1_1} = \frac{a_{1_1} + b_{1_1}}{2} = \frac{1}{4} \\ c'_{1_1} = \frac{a'_{1_1} + b'_{1_1}}{2} = \frac{1}{2} \end{cases} \Rightarrow \begin{cases} X_{1_1} = (0.25, 0.5) \\ UB_{1_1} = 0.251 \\ LB_{1_1} = 0.051 \end{cases}$$

2^{ieme} Cas: $x \in [a_{1_2}, b_{1_2}] \times [a'_{1_2}, b'_{1_2}] = [\frac{1}{2}, 1] \times [0, 1]$

(a) Calcul de c_{1_2} , c'_{1_2} , X_{1_2} , UB_{1_2} et LB_{1_2} :

$$\begin{cases} x_1 \in [\frac{1}{2}, 1] \\ x_2 \in [0, 1] \end{cases} \Rightarrow \begin{cases} c_{1_2} = \frac{a_{1_2} + b_{1_2}}{2} = \frac{3}{4} = 0.75 \\ c'_{1_2} = \frac{a'_{1_2} + b'_{1_2}}{2} = \frac{1}{2} = 0.5 \end{cases} \Rightarrow \begin{cases} X_{1_2} = (0.75, 0.5) \\ UB_{1_2} = 0.75 \\ LB_{1_2} = 0.94 \end{cases}$$

3. Calcul de LB_1 et UB_1 :

$$\begin{cases} UB_1 = \min\{UB_{1_1}, UB_{1_2}, UB_0\} = -0.251 \\ LB_1 = \min\{LB_{1_1}, LB_{1_2}\} = 0.051 \end{cases}$$

4. Test d'arrêt:

$UB_1 - LB_1 = -0.2 < \epsilon$, donc $UB_k = \min f(X)$ et $X^k = \hat{X}$.

Tableau représentatif des résultats du deuxième exemple.

| K | x^{K_1} | x^{K_2} | UB_{K_1} | UB_{K_2} | UB_K | LB_{K_1} | UB_{K_2} | UB_K |
|-----|-------------|-------------|------------|------------|--------|------------|------------|--------|
| 0 | (0.5, 0.5) | | 0.538 | | 0.538 | 0.220 | | 0.220 |
| 1 | (0.25, 0.5) | (0.75, 0.5) | -0.251 | 0.75 | -0.251 | 0.051 | 0.94 | 0.051 |

3.0.3 Conclusion

Dans ce chapitre on a traité deux exemples numériques de fonctions à deux variables sur un hyper rectangle de \mathfrak{R}^2 , en appliquant l'algorithme α BB

Chapitre 4

Programmation dans LINGO

4.1 Introduction

La programmation est un ensemble d'outils et de technique permettant de résoudre des problèmes mathématiques par ordinateurs, elle sert a trouvé une solution optimal de n'importe quel type de problèmes.

Le processus de résoudre un problème mathématiques exige un grand nombre de calculs donc il est mieux de l'exécuter par machine . Pour cela on a choisit le logiciel LINGO qui est spécifié pour résoudre des problèmes d'optimisations (linéaire,non linéaire,convexe ,non convexe,..etc).

Il comporte un éventail de commandes qui peuvent être appelé à tout moment .

4.1.1 Présentation du logiciel

LINGO est livré avec un jeux de solveurs pour l'optimisation .En entrée, ce solveur LINGO doit recevoir un modèle mathématique et il possède quatre solveurs qu'il utilise afin de résoudre les différents types de modèles :

- Solveur direct.
- Solveur linéaire.
- Solveur non linéaire.
- Méthode de type séparation et évaluation.

De plus, LINGO est:

- Un moyen pour confirmer que l'optimum trouvé est l'optimum global.
- Un moyen amélioré pour résoudre beaucoup de type de problèmes .
- Un moyen de décomposition si un modèle contient des sous-modèles.
- Possible de résoudre les problèmes plus rapidement .

Les fonctions utilisées dans un modèle de LINGO sont @FOR-utilisée pour pro-

duire des contraintes.

@SUM-calcul de la somme.

@MAX-recherche le maximum.

@MIN-recherche le minimum.

Type de variables dans LINGO

Toutes les variables dans un modèle LINGO sont considérées non négatives et continues. Les fonctions variables d'un modèle de LINGO sont:

@GIN-toute valeur positive de nombre entier.

@BIN-une valeur binaire (0 ou 1).

@FREE-toute valeur positive ou négative réelle .

@BND-toute valeur bornée par des limites indiquées .

Sa forme générale pour la déclaration d'une variable x en utilisant la fonction @BND qui inclut les bornes inférieures et supérieures est:

@BND(borne inférieure ,x,borne supérieure)

Pour savoir plus sur ce logiciel, regardez les références:[4],[6],[8]

4.1.2 Interface du logiciel

Lors de lancement de LINGO la barre des menus et une fenêtre pour saisir le modèle du problème à résoudre apparaît :

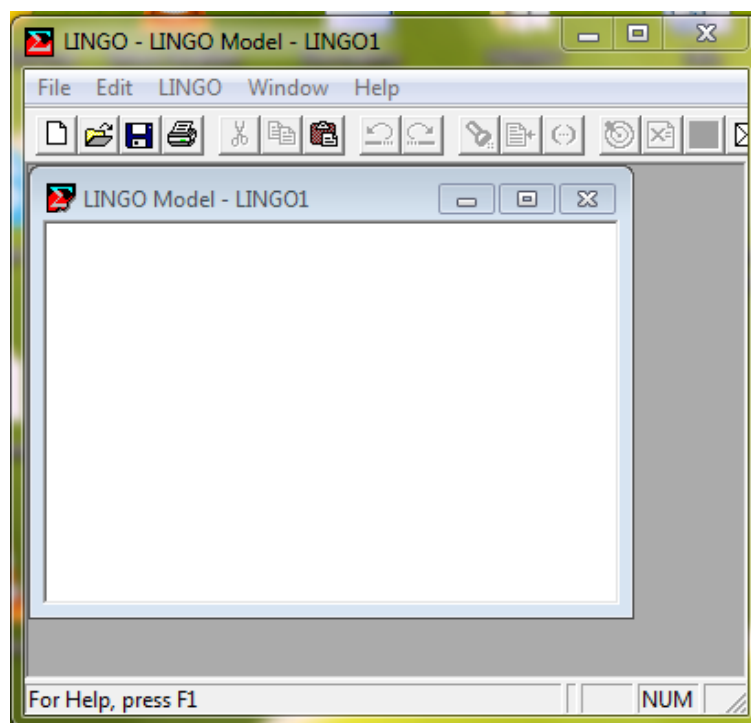


FIG. 4.1 – xxx

On peut sélectionner les options à suivre pour résoudre un modèle en cliquant sur LINGO options . Dans la fenêtre suivante on sélectionne "global solver".

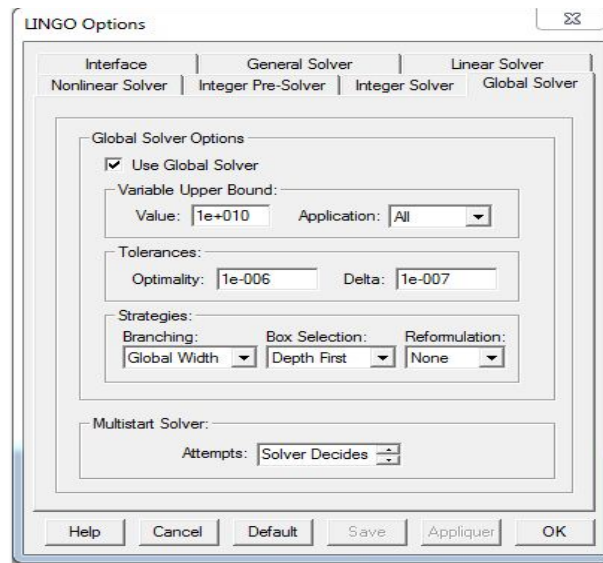


FIG. 4.2 –

Ensuite on fait rentrer le modèle de problèmes à résoudre. Pour ouvrir un modèle enregistré en clique sur "open" dans la barre des menus ,la fenêtre suivante apparaît :

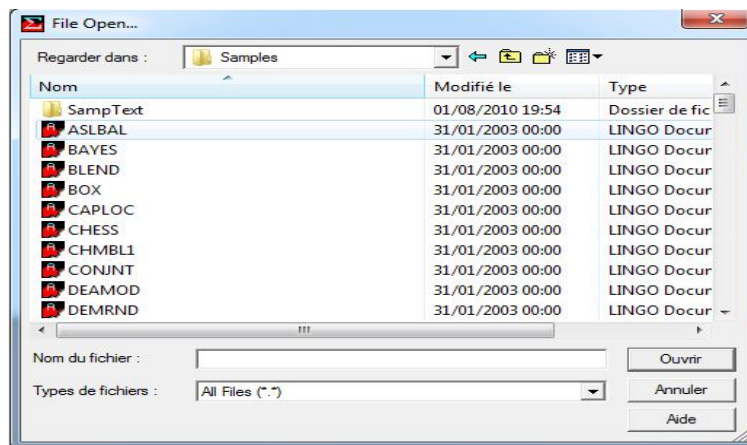
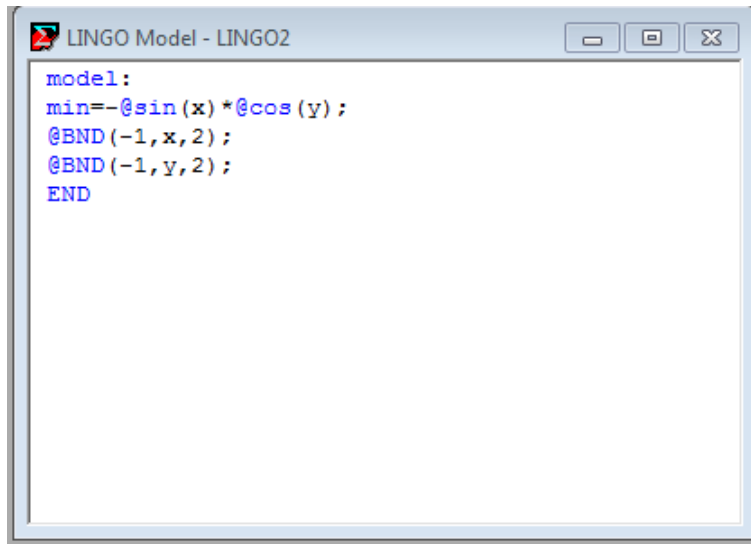


FIG. 4.3 –

On sélectionne le model a ouvrir en cliquant sur "ouvrir" dans la fenêtre suivante le modèle sélectionné s'ouvre dans la fenêtre "LINGO model":



```
model:
min=-@sin(x)*@cos(y);
@BND(-1,x,2);
@BND(-1,y,2);
END
```

FIG. 4.4 –

Une fois le modèle est entré il peut être résolu en cliquant sur le bouton "solve" sur la barre des outils.

En sélectionnant LINGO/solve de menu, ou en utilisant un raccourci du clavier "ctrl+s". Il annonce les erreurs rencontrées, la meilleure manière pour avoir des informations sur une erreur c'est de consulter la section "Error Messages".

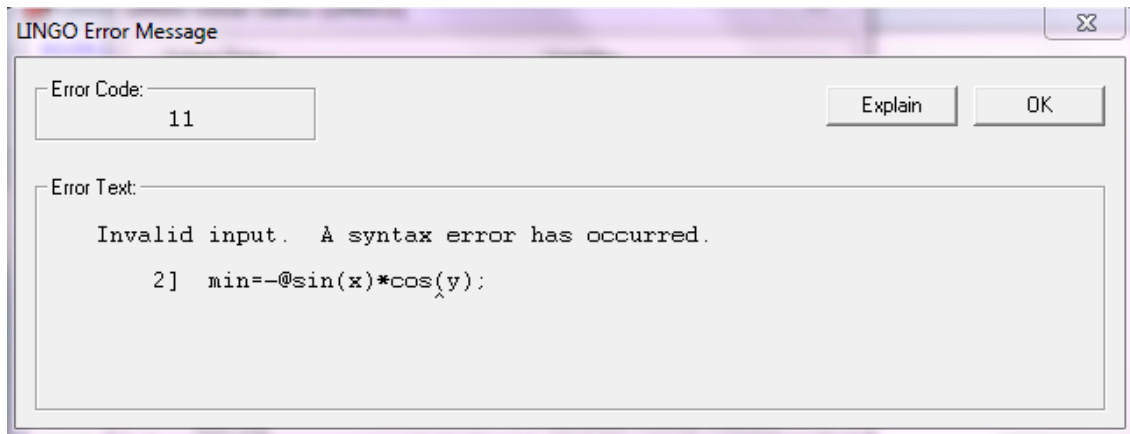


FIG. 4.5 –

Si aucune erreur n'est signalée alors la fenêtre suivante apparaît:

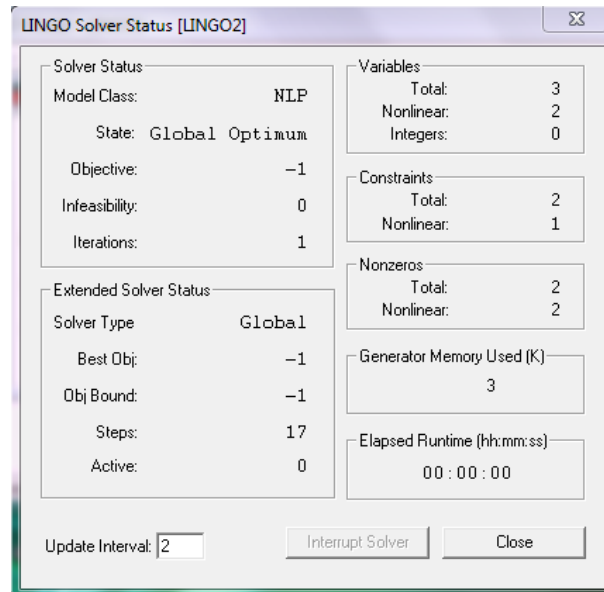


FIG. 4.6 –

Cette fenêtre fournit des informations sur le nombre de variables non linéaires ,le type de la solution obtenue (locale ou globale) la valeur de la fonction objectif ,le nombre d'itérations requise pour résoudre le modèle .

En fermant cette fenêtre on peut alors regarder la fenêtre de rapport de solution .

4.1.3 Résolution des exemples traités dans le chapitre 3 sur LINGO avec la méthode Branch-and-Bound

Exemple 4.1. Soit à résoudre le problème suivant:

$$(P) \begin{cases} \min f(x) \\ X \in H \end{cases}$$

Où $f(x_1, x_2) = x_1^3 - x_1 x_2^2$

$X = (x_1, x_2) \in H, H = [0,1] \times [0,1], \alpha = 1.25, \epsilon = 10^{-5}$

Le modèle d'exécution sur LINGO est le suivant: MODEL :

MIN = $x_1^3 - x_1 x_2^2$

@*BND*(0, x_1 , 1)

@*BND*(0, x_2 , 1)

END

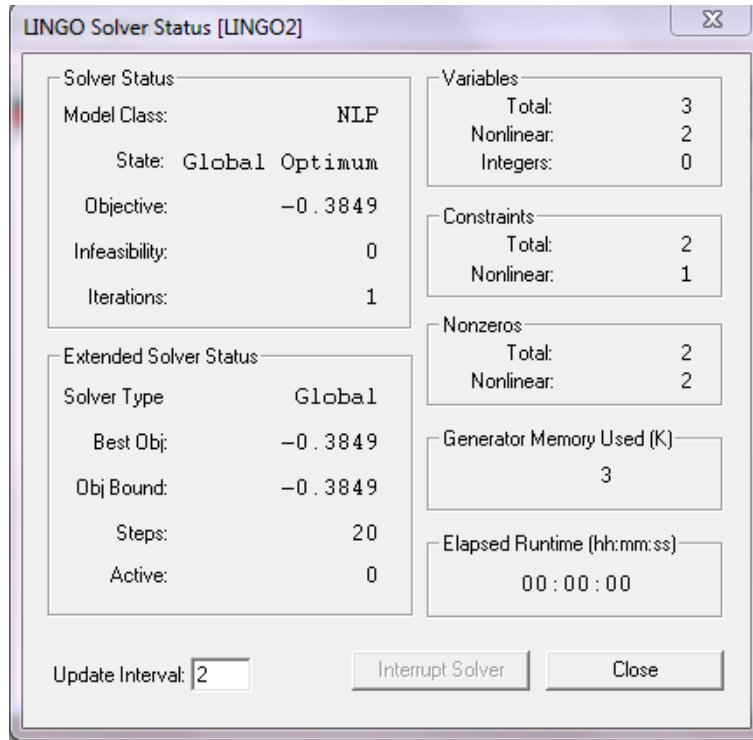


FIG. 4.7 – Résultat de l'exécution du model

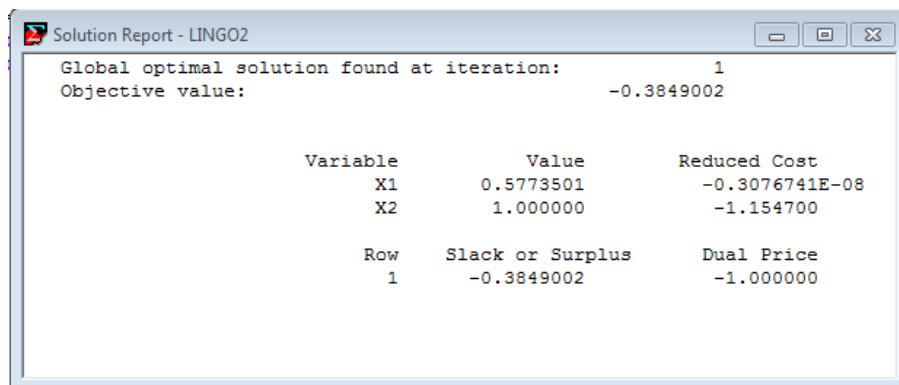


FIG. 4.8 – Résultat de l'exécution du model

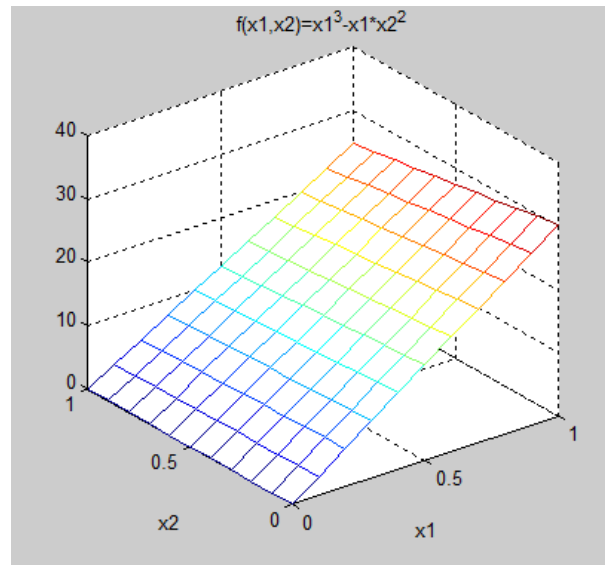


FIG. 4.9 – Le graphe de la fonction

Exemple 4.2. $f(x_1, x_2) = x_1 \cos x_2 + x_2 \sin x_1$ avec $(x_1, x_2) \in H, H = [0, 1] \times [0, 1], \alpha = 0.635, \epsilon = 10^{-5}$

Le modèle d'exécution sur LINGO est le suivant:

MODEL :

MIN = $x_1 \cos x_2 + x_2 \sin x_1$

@BND(0, x₁, 1)

@BND(0, x₂, 1)

END

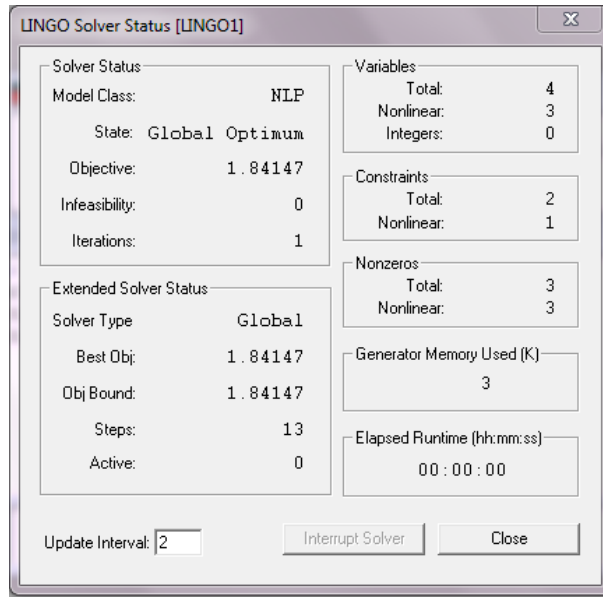


FIG. 4.10 – Résultat de l'exécution du modèle

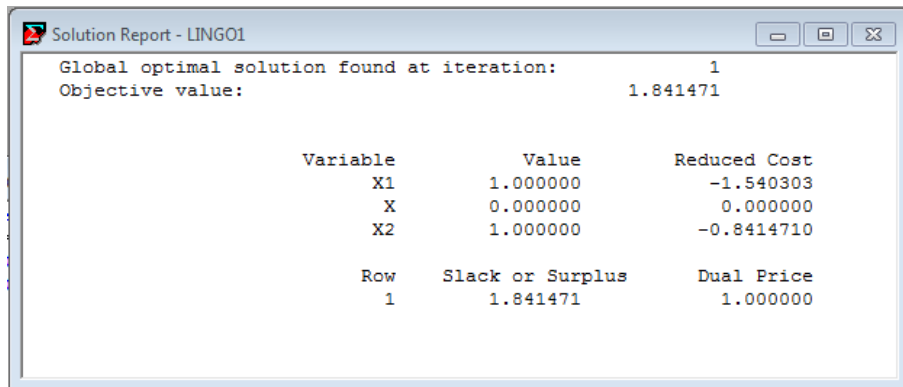
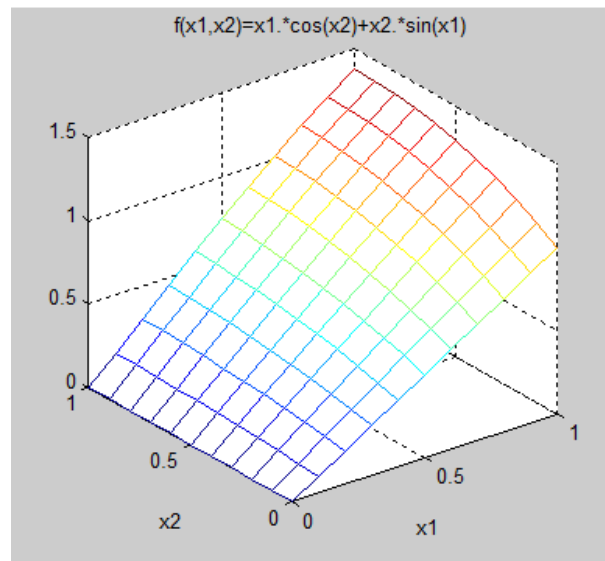


FIG. 4.11 – Résultat de l'exécution du modèle

FIG. 4.12 – *Le graphe de la fonction*

4.1.4 Conclusion

On a vu dans ce chapitre comment insérer un model d'execution d'un problème sur LINGO dans le cas des problèmes traités par la méthode Branch and Bound.

Lingo fait appel au "solveur global" après l'execution on a directement le minimum global et la valeur optimale de la fonction objectif

conclusion et perspectives

Nous nous sommes intéressée dans notre travail à la résolution des problèmes d'optimisation globale des fonctions deux fois différentiable non linéaires, non convexes sur un intervalle en tenant compte de leurs structures telles que la linéarité et la convexité.

Nous avons présenté une méthode de résolution précisément la méthode déterministe qui est beaucoup plus efficace dans la détermination des minimums globaux .

La méthode Branch-and-Bound a été utilisée dans plusieurs domaines d'optimisation comme l'optimisation combinatoire, l'optimisation semi-infini et l'optimisation quadratique ainsi que l'optimisation globale.

Pour montrer l'efficacité de l'algorithme α BB qu'on a étudié dans le deuxième chapitre ,on a traité quelques exemples numériques avec une constante α

La vérification des résultats a été faite sur le logiciel LINGO en programmant quelques exemples de problèmes de fonctions à deux variables.

On a comparé les résultats obtenus dans le troisième chapitre et ceux trouvés dans le quatrième chapitre. On a remarqué que la valeur minimale trouvée en utilisant le paramètre α est plus petite que celle calculée en utilisant le solveur LINGO.

Comme perspectives, il est souhaitable de traiter des problèmes d'optimisation globale par intervalle pour les fonctions deux fois différentiable en utilisant un langage de programmation ,cela dans le but d'accélérer la convergence de l'algorithme.

Bibliographie

- [1] Adjman ,IP.Androulakis CD.MaranasC.A Floudas.” A Global optimization Method, α BB For process Design” .
- [2] Mr AMIROU Ahmed . ”Etude et implémentation des méthodes d’optimisation.Semi-infinie et Branch and Bound ” .Mémoire de magister,UMMTO(2006).
- [3] ANDROULAKIS ,MARANAS and FLOUDAS ,” α BB :A Global Optimization Method for General Constrained Non convex Problems”.(1995).
- [4] Melle BEKRI Houria . ” Optimisation globale avec application pour les fonctions de holder ” . Mémoire de magister,UMMTO(2012).
- [5]Melle BELHACENE Saliha. ” Résolution de problème de programmation semi-infinie et introduction à l’optimisation globale”. Mémoire de magister,UMMTO(2005).
- [6] Melle BOUZEBODJA Tassadit .”Optimisation globale avec application pour les fonctions lipschiziennes”.Ingénieur d’état en recherche opérationnelle .
- [7] Claire S.Adjiman and Christodoulos A.Floudas ,Rigorous ”convex Understimators for General Twice-Differentiable Problems”.
- [8] Mr HAMADI Hamza . ” Etude de Branch and bound avec mise en place d’heuristiques pour coupe de branche ,Application :Programmation de la méthode B-and-B pour la résolution des PLNE .
- [9] Jaques Gauvin.”Leçons de programmation mathématique”,éditions de l’école polytechnique de Montréal(1995).
- [10]JORDAN Ninin . ” Optimisation globale basée sur l’analyse d’intervalle ” Relaxation affine et limitation de la mémoire .Thèse de doctorat a l’université de Toulouse (08 decembre 2010).
- [11]MESSINE Frédéric .Méthode d’optimisation globale basée sur l’analyse d’intervalle pour la résolution de problèmes avec contraintes.Thèse de doctorat ,INPDT ,UPS ,31062 Toulouse,26 septembre (1997).

[12] MICHEL Minoux ."Programmation mathématique ,théorie et algorithme ".Professeur des universités UFR d'ingénierie,laboratoire d'informatique de l'université Pierre et Marie Curie(Paris 6).2éme édition (2006).

[13]Mr OUANES .Mohand ."Optimisation semi-infinie et Optimisation Globale,Théorie et Algorithmes".Thèse d'état,UMMTO,(2006).

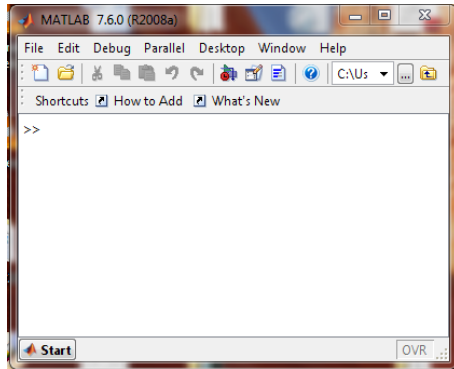


FIG. 4.13 –

annexe

Le nom de MATLAB dérivé du terme MATRIX LABORATORY .Il s'agit d'un outil de développement pour des problèmes scientifiques et plus généralement pour tous les domaines ou des calculs numériques importants doivent être faits.

MATLAB travaille essentiellement avec des ensembles rectangulaires carrés (matrices)de données, les éléments peuvent être réels ou complexes.

Au démarrage de MATLAB la fenêtre de commande s'ouvre .Elle permet d'exécuter immédiatement une instruction ou un programme.

dans cette fenêtre .Chaque instruction est précédée du signe>>.

On a utilisé MATLAB pour créer les graphes des fonctions traitées dans le chapitre3. Les commandes utilisées sont les suivantes :

Plot,axis,Meshgrid,Mesh,Xlabel,Ylabel,Zlabel,TITLE,Figure.Axis.

Utilisation:

Plot:tracer le graphe 2D de la fonction.

Meshgrid:déclaration des variables.

Mesh: tracer le graphe 3D de la fonction.

Xlabel: ajoute une légende pour l'axe des abscisses.

Ylabel:ajouté une légende pour l'axe des ordonnées.

Zlabel:ajoute une légende pour l'axe des Z.

TITLE : ajoute un titre.

Figure:créer une nouvelle fenêtre.

Axis :Modifie les échelles des axes.

exemple

```
[x1,x2]=Meshgrid (0:0.1:1,0:0.1:1);  
F=x1.*cos(x2)+x2.*sin(x1);  
Mesh(x1,x2,F);  
Xlabel('x1');  
Ylabel('x2');  
Title('f(x1,x2)=x1.*cos(x2)+x2.*sin(x1)');  
Axis([x1min,x1max,x2min,x2max]);
```