

الجمهورية الشعبية الديمقراطية الجزائرية

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE

SCIENTIFIQUE

جامعة عمّار تليجي بالأغواط



Université Amar Téliidji Laghouat

Faculté De Technologie

Département d'électronique

Polycopier de Cours

KAMRI DJEKIDEL

Introduction à l'Optimisation

Programmation Mathématique

KAMRI DJEKIDEL

2022

Page vierge

Page vierge

Page vierge

Page vierge

Page vierge

Page vierge

Page vierge

Page vierge

Avant Propos

Ce document constitue le support de cours du module Optimisation pour master Automatique & Informatique Industrielle et pourrait être utile pour tous les étudiants des filières du domaine technique. L'objectif de ce polycopié est de fournir à l'étudiant les bases des méthodes d'optimisations ainsi que l'implémentation algorithmiques des techniques de résolution numériques utilisées (issues de l'analyse numérique). En engineering, Nous nous attachons plutôt aux résultats mathématiques bien établis qui représentent l'outil de base pour comprendre les techniques d'optimisation et leurs implémentations informatiques sans s'enfoncer dans les détails et les démonstrations mathématiques. Les prés requis sont essentiellement l'algèbre linéaire, l'analyse numérique matricielle et le calcul différentiel. Ce cours ne représente qu'une introduction à une théorie multidisciplinaire qui pourrait nécessiter plusieurs centaines de pages pour être présenter.

BIBLIOGRAPHIE DE BASE

- Aharon ben_tal and Arkadi nemirovski, convex optimization in engineering "modeling_analysis_algorithms", technion – israel institute of technology, 1998, 623p.
- Guy cohen, convexité et optimisation, école d'ingénieur. 1999-2001 école nationale des ponts et chaussées, 2000, 139p.
- Megretski, optimization methods in analysis and design of linearized systems, 2005, 176p.
- Yadolah dodge, Sylvie gonano-weber, Jean-pierre renfer, optimisation appliquée, éditions edes de neuchâtel, 2004, 343p.
- Franck jedrzejewski. introduction aux méthodes numériques, springer 2005, 284p.
- Stephen boyd, lieven Vandenberghe, convex optimization, cambridge university press, 2004, 730p.
- Stephen boyd, Laurent El-ghaoui, eric feron, and venkataramanan balakrishnan; linear matrix inequalities in system and control theory, siam studies in applied mathematics, 1994, 205p.
- Marvin .marcus and henryk minc, matrix theory and matrix inequalities, 1964, 198p.
- David a. Harville, matrix algebra: exercises and solutions, springer, 2001, 306p.
-

Quelques Sources électroniques

<https://www.ljll.math.upmc.fr/~documents/Cours-Optimisation.pdf>

<http://math.univ-lyon1.fr/~ciuperca/cours-optim-MISAF.pdf>

https://www.math.sciences.univ-nantes.fr/Résumé_d'Optimisation

http://perso.eleves.ens-rennes.fr/~enseignement/Cours_d'optimisation_ENSAI_Rennes

http://dumas.perso.math.cnrs.fr/~MINT-2018-TO1/Techniques_d'optimisation

https://www.math.sciences.univ-nantes.fr/Résumé_d'Optimisation

Table des matières

Introduction générale.....	I
Notions préliminaires.....	1
Espaces vectoriels.....	1
Ensembles convexes	2
Suites numériques dans un espace vectoriel normé.....	6
Généralités sur les matrices.....	7
Rappel de Calcul différentiel.....	14
Développements limités:.....	19
Calcul numérique sur ordinateur.....	21
o Calcul en virgule flottante:.....	21
o Epsilon machine:.....	22
o Erreurs et précision:.....	22
o Méthodes itératives.....	23
Chapitre 1 : Introduction à Optimisation	27
Introduction.....	28
1.1. Éléments introductifs à l'optimisation.....	29
1.2. Formulation mathématique d'un problème d'optimisation	32
1.3. Méthode des moindres carrés	34
1.4. Exemples d'optimisation en pratique:.....	36
1.4.1. Exemple 1:.....	36
1.4.2. Exemple 2.....	37
1.4.3.Exemple 3.....	38
1.4.4. Exemple 4: Problème de transport.....	38
1.4.5. Exemple 5: Modélisation de données expérimentales (Identification).....	38
1.5. Nature des contraintes/qualification des contraintes:.....	43
Chapitre 2 : Optimisation sans contraintes.....	47
2.1. Conditions d'existence et d'unicité.....	47
2.2. Conditions d'optimalité:.....	48
2.3. Cas quadratique.....	50
2.4. Résolution Algorithmique:.....	52
2.5. Méthode des approximations successives:.....	52
2.6. Méthodes du gradient:.....	52
2.6.1. Sélection d'un pas ρ_k adéquat.....	54
2.6.2. Gradient à pas optimal.....	55
2.6.3. Méthodes du gradient conjugué	56
2.7. Méthode de Newton:.....	56
2.8. Méthodes quasi-Newton.....	59

Chapitre 3 : Optimisation avec contraintes.....	62
3.1. Conditions d'optimalité:.....	63
3.2. Multiplicateur de Lagrange/ contrainte d'égalité	64
3.3. Multiplicateur de Lagrange/ contrainte d'inégalité	69
3.4. Multiplicateur de Lagrange en présence de contraintes d'égalité et d'inégalité:...	71
3.5. Méthodes de Programmation Quadratique Successive (SQP).....	73
3.6. Introduction à la théorie de la dualité:.....	75
3.7. Méthodes de pénalisation	77
Chapitre 4 : Programmation linéaire.....	79
4.1. Ensemble-Solution d'inéquation ou système d'inéquations linéaires:.....	80
4.2. Programme Linéaire (forme canonique).....	82
4.3. Programme Linéaire (forme standard ou normale).	84
4.4. Caractérisation des solutions:.....	84
4.5. Recherche de solution optimale:.....	87
4.6. La méthode simplexe du tableau.....	88
4.7. Exemple d'application:.....	89
4.8. Dualité minimum/maximum.....	93
Annexe Inéquations Matricielles Linéaires (LMI).....	96

Notions Préliminaires

Rappel mathématique

Les espaces de Hilbert sont un bon cadre pour la formulation de problèmes d'optimisation rencontrés par l'ingénieur. L'espace \mathfrak{R}^n muni de son produit scalaire classique est l'exemple le plus connu. Le fait de ne pas se limiter à la dimension finie permet par exemple d'aborder les problèmes de commande optimale en temps continu où les grandeurs appropriées sont des fonctions du temps que l'on peut munir d'une norme.

Si en optimisation, on s'intéresse plus particulièrement aux fonctions: $f: \Omega \subset \mathfrak{R}^p \rightarrow \mathfrak{R}^q$ ($p, q \in \mathbb{N}^$), alors il faudrait bien étudier les structures du domaine Ω , car ce dernier est aussi important que la fonction elle-même.*

Espaces vectoriels

Soit E un ensemble. On dispose sur cet ensemble d'une opération (notée additivement) et on dispose par ailleurs d'une application $K \times E \rightarrow E$ qui à tout couple (λ, x) associe λx . On dit que E est un espace vectoriel lorsque:

- E est un groupe commutatif (pour l'addition)
- pour tout vecteur x de E , $1 \cdot x = x$ (1 désignant le neutre de la multiplication de K).
- pour tous $\lambda, \mu \in K$ et pour tout vecteur x de E , $(\lambda, \mu)x = \lambda(\mu x)$
- pour tous $\lambda, \mu \in K$ et pour tout vecteur x de E , $(\lambda + \mu)x = \lambda x + \mu x$
- pour tous $\lambda \in K$ et pour tout vecteurs x, y de E , $\lambda(x + y) = \lambda x + \lambda y$

Espaces métriques

Un espace vectoriel E (réel) est un espace métrique lorsqu'il est muni d'une *distance* d , c'est à-dire d'une application de $E \times E$ dans \mathfrak{R} vérifiant :

- $\forall x \in E, \forall y \in E$ alors $d(x, y) \geq 0$ et $d(x, y) = 0 \Leftrightarrow x = y$
- $\forall x \in E, \forall y \in E$ alors $d(x, y) = d(y, x)$
- $\forall x \in E, \forall y \in E, \forall z \in E$ alors $d(x, z) = d(x, y) + d(y, z)$

Norme des Espaces vectoriels

Soit E un espace vectoriel sur \mathfrak{R} (on utilisera en général $E = \mathfrak{R}^n$). On appelle norme sur E une application

$$\begin{cases} E \rightarrow \mathfrak{R}^+ \\ x \rightarrow \|x\| \end{cases}, \text{ vérifiant les propriétés suivantes:}$$

- $\forall x \in E, \forall y \in E$ alors $\|x\| > 0$ et $\|x\| = 0 \Leftrightarrow x = 0$
- $\forall x \in E, \forall y \in E$ alors $\|x + y\| \leq \|x\| + \|y\|$
- $\forall x \in E, \forall y \in E$ alors $\|x - y\| \geq \|x\| - \|y\|$
- $\forall x \in E, \forall \alpha \in K$, alors $\|\alpha x\| = |\alpha| \|x\|$

Espaces normés

Un espace normé est un espace métrique dont la distance est définie à partir d'une norme. Plus précisément, la distance $d(x, y)$ est définie par

$$d(x, y) = \|x - y\|$$

Autrement, Un espace vectoriel normé est un espace vectoriel sur \mathfrak{R} muni d'une norme.

Définition : (boule ouverte, fermée, sphère)

Soit $(E, \|\cdot\|)$ un espace vectoriel normé et soient a un point de E , r un réel positif:

- $\bar{B}(a, r) = \{x \in E; \|x - a\| \leq r\}$ est appelé boule fermée de centre a et de rayon r .
- $B(a, r) = \{x \in E, \|x - a\| < r\}$ est appelé boule ouverte de centre a et de rayon r .
- $S(a, r) = \{x \in E; \|x - a\| = r\}$ est appelé Sphère de centre a et de rayon r .

Dans le cas où $a = 0$ (vecteur nul) et $r = 1$ on a ce qu'on appelle les boules ou sphères unités.

Définition : (ensemble borné)

Soit $(E, \|\cdot\|)$ un espace vectoriel normé et soit $K \subset E$; On dit que K est borné si on peut trouver une boule (ouverte ou fermée) qui contient tous les points de K .

Définition :

Pour un ensemble $S \subseteq \mathbb{R}^n$, son intérieur est défini par :

$$\text{Int}(S) = \{x \in S \mid \exists \varepsilon > 0; B(x, \varepsilon) \subseteq S\}$$

Définition : (ensemble ouvert)

Soit $(E, \|\cdot\|)$ un espace vectoriel normé et soit $K \subset E$; On dit que K est un ouvert si tout point x de K est le centre d'une boule ouverte de rayon non-nul $B(x, r) \subset K$.

Ainsi un ensemble $S \subseteq \mathbb{R}^n$ est ouvert si il est équivalent à son intérieur : S ouvert $\Leftrightarrow S = \text{Int}(S)$.

Définition : (ensemble fermé)

Un ensemble $S \subseteq \mathbb{R}^n$ est dit fermé si tout point extérieur de S possède un voisinage disjoint de S : S fermé $\Leftrightarrow \forall x \notin S, \exists \varepsilon > 0 : B(x, \varepsilon) \cap S = \emptyset$.

Par exemple:

Une boule fermée est un fermé.

Une boule ouverte est un ouvert.

Propriétés: (ensemble ouvert / fermé)

1. toute union finie ou infinie d'ouverts de E est un ouvert.
2. toute intersection FINIE d'ouverts de E est un ouvert.
3. toute union FINIE de fermés de E est un fermé.
4. toute intersection finie ou infinie de fermés de E est un fermé.
5. les ensembles à la fois ouverts et fermés de E sont ; \emptyset et E , et si ce sont les seuls on dira que l'espace est CONNEXE
6. un ensemble fini de points de E est fermé.

Définition: (ensemble compact)

Soit $(E, \|\cdot\|)$ un espace vectoriel normé et soit $K \subset E$; On dit que K est compact s'il est fermé et borné dans E .

Définition: (ensemble convexe)

Un ensemble $S \subseteq \mathbb{R}^n$ est dit convexe, si le segment joignant n'importe quels deux points de S est complètement dans S .

$$S \text{ convexe} \Leftrightarrow \theta x_1 + (1 - \theta)x_2 \in S, \forall x_1, x_2 \in S, 0 \leq \theta \leq 1.$$

D'une manière générale, un ensemble $S \subseteq \mathbb{R}^n$ est dit convexe, si quelque soit les points x_1, x_2, \dots, x_k de S , leur combinaison convexe est dans S .

$$\sum_{i=1}^k \theta_i x_i \in S, \theta_i \geq 0 \text{ et } \sum_{i=1}^k \theta_i = 1$$

Propriété 1 :

L'intersection d'un nombre quelconque d'ensembles convexes est un ensemble convexe.

Propriété 2 :

L'image d'un ensemble convexe par un opérateur linéaire est un ensemble convexe.

Définition : (Coque)

La coque d'un ensemble convexe $S \subseteq \mathbb{R}^n$, noté $co(S)$ est l'ensemble convexe qui contient S et ne contient aucun ensemble qui contient S . elle peut être exprimée par :

$$\left\{ \sum_{i=1}^k \theta_i x_i \mid x_i \in S; \theta_i \geq 0, \sum_{i=1}^k \theta_i = 1 \right\}$$

Définition : (fonction convexe)

Soit $S \subseteq \mathbb{R}^n$ un ensemble convexe, et soit $f : S \rightarrow \mathbb{R}$ une fonction, on dit que f est convexe si et seulement si :

$$\forall x_1, x_2 \in S \text{ et } \forall \theta \in (0,1): f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2)$$

Définition : (fonction affine)

Une fonction $f : S \rightarrow \mathbb{R}^m$, $S \subseteq \mathbb{R}^n$, est dite affine s'il existe $\alpha \in \mathbb{R}^m \times \mathbb{R}^n$ et $\beta \in \mathbb{R}^m$ tel que pour tout $x \in S$: $f(x) = \alpha x + \beta$.

Si de plus S est convexe alors f est convexe (mais pas strictement).

Définition : (fonction affine par morceaux)

Une fonction $f : S \rightarrow \mathbb{R}^m$, $S \subseteq \mathbb{R}^n$ est dite affine par morceaux (PWA), s'il existe une partition $\{S_i\}_{i=1}^p$ de S avec $p \in \mathbb{N}$, des matrices $H_i \in \mathbb{R}^{m \times n}$ et des vecteurs $k_i \in \mathbb{R}^m$, $i = 1, \dots, p$, telle que :

$$f(x) = \begin{cases} H_1 x + k_1 & \text{si } x \in S_1, \\ H_2 x + k_2 & \text{si } x \in S_2, \\ \vdots & \\ H_p x + k_p & \text{si } x \in S_p, \end{cases}$$

Définition : (forme quadratique)

Un ensemble quadratique est de la forme suivante :

$$S = \{x \in \mathbb{R}^n \mid x^t Q x + 2h^t x + k \leq 0\}$$

$$= \left\{ x \in \mathbb{R}^n \mid \begin{pmatrix} x \\ 1 \end{pmatrix}^t \begin{pmatrix} Q & h \\ h^t & k \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} \leq 0 \right\}$$

$Q \in \mathbb{R}^n \times \mathbb{R}^n$, $h \in \mathbb{R}^n$ et $k \in \mathbb{R}$

Définition : (hyperplan)

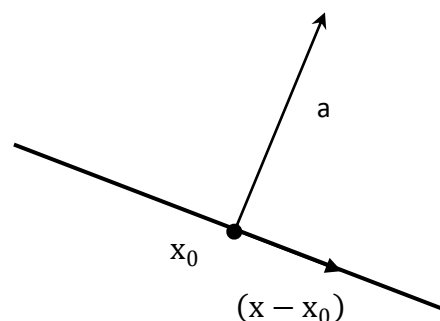
Un hyperplan dans \mathbb{R}^n est un ensemble de la forme :

$$H = \{x \in \mathbb{R}^n \mid a^t x = b\}, \quad a \in \mathbb{R}^n, a \neq 0_n, b \in \mathbb{R}$$

Ou équivalamment :

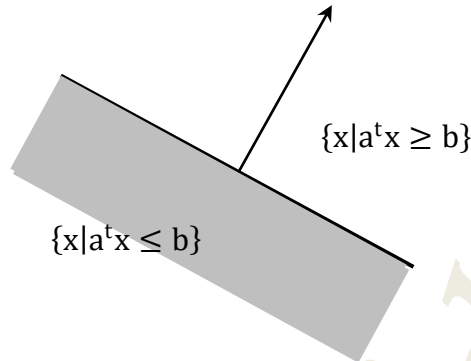
$$H = \{x \in \mathbb{R}^n \mid a^t (x - x_0) = 0\}, \quad b = a^t x_0$$

Où a est le vecteur normal au plan.



Un hyperplan est un ensemble affine (dimension $(n-1)$) et convexe, il divise l'espace R^n en deux demi-espaces ouverts.

$$H^- = \{x \in R^n | a^t x - b < 0\} \quad \text{et} \quad H^+ = \{x \in R^n | a^t x - b > 0\}$$



L'intersection d'un nombre fini d'hyperplan est un ensemble affine :

$$A = \{x \in R^n | Ax = b\}, \quad A \in R^{m \times n}, b \in R^m$$

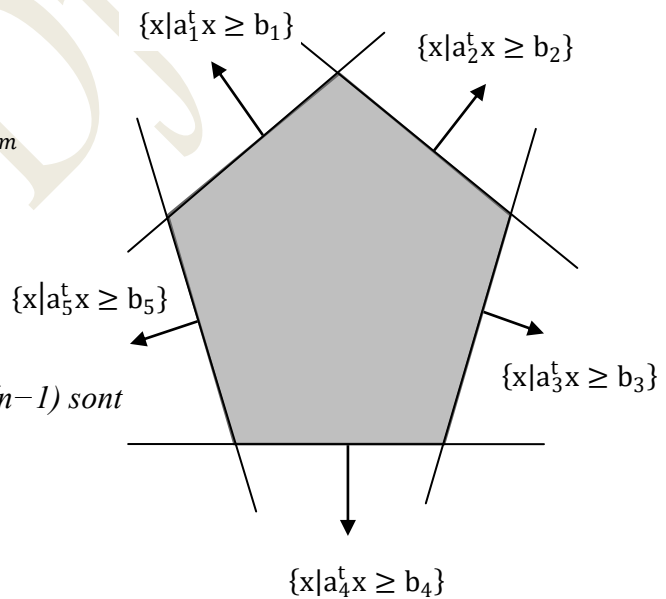
Définition : (Polyèdre)

L'intersection définie par un nombre fini de demi-espaces est un polyèdre (ensemble convexe)

$$\begin{aligned} P &= \{x \in R^n | a_i^t x - b_i < 0\}, \quad i = 1, 2, \dots, m \\ &= \bigcap_{i=1}^m [a_i^t x - b_i < 0] \\ &= \{x \in R^n | Ax - b < 0\}, \quad A \in R^{m \times n}, b \in R^m \end{aligned}$$

D'autre part : $\bar{P} = \bigcap_{i=1}^m [a_i^t x - b_i \leq 0]$

Où $F = P \cap \{x \in R^n | a_i^t x = b_i\}$ est appelée face du polyèdre ; faces de dimension 0, 1 et $(n-1)$ sont appelées « vertiges », « edges » et « facets ».



Définition : (Polytope)

Un polytope est un polyèdre fermé et borné où chaque point peut être représenté par une combinaison convexe de ses sommets. Il représente donc la coque de ses points sommets, cette représentation est appelée v -représentation qui a l'avantage d'être décrite par des hyperplans.

$$\mathcal{P} = co(\{v_1, v_2, \dots, v_{n_v}\}),$$

Beaucoup d'algorithmes d'optimisation sont basés sur la représentation polytopique ou ellipsoïdale des régions d'espace.

Suites numériques dans un espace vectoriel normé

Définition d'une suite:

On appelle suite dans un espace vectoriel normé E toute application:

$$\begin{cases} \mathbb{N} \rightarrow E \\ n \rightarrow x_n \end{cases} \quad \text{et l'a note : } (x_n)_{n \in \mathbb{N}}$$

Définition (suite bornée):

Soit $(x_n)_{n \in \mathbb{N}}$ une suite de E muni de la norme $\|\cdot\|$, $(x_n)_{n \in \mathbb{N}}$ est dite bornée si et seulement si l'ensemble $\{x_n, n \in \mathbb{N}\}$ est borné. Autrement dit, il existe $M > 0$ tel que pour tout $n \in \mathbb{N}$, $\|x_n\| < M$.

L'ensemble des suites bornées dans un espace vectoriel normé est un espace vectoriel.

Définition (suite convergente):

Soit $(x_n)_{n \in \mathbb{N}}$ une suite de E muni de la norme $\|\cdot\|$, On dit que la suite $(x_n)_{n \in \mathbb{N}}$ converge dans $(E, \|\cdot\|)$, si et seulement si, il existe $l \in E$, telque pour tout $\varepsilon > 0$, il existe $m \in \mathbb{N}$ telque pour tout $n \geq m$, on a $\|x_n - l\| < \varepsilon$.

La limite l de la suite $(x_n)_{n \in \mathbb{N}}$ définie ci-dessus est unique et l'ensemble des suites convergentes dans un espace vectoriel normé est aussi un espace vectoriel.

Définition (suite convergente dans un ensemble fermé)

Soit $S \subseteq E$, S est fermé si et seulement toute suite de points de S qui converge possède une limite qui appartient à S .

Définition (suite de CAUCHY)

Soit $(x_n)_{n \in \mathbb{N}}$ une suite de E muni de la norme $\|\cdot\|$, On dit que la suite $(x_n)_{n \in \mathbb{N}}$ converge dans $(E, \|\cdot\|)$, si et seulement si, il existe $m \in \mathbb{N}$, telque pour tout $\varepsilon > 0$ et $n, k > m$,

on a $\|x_n - x_k\| < \varepsilon$.

Si une suite d'un espace vectoriel normé est convergente alors elle est de CAUCHY et l'espace vectoriel normé est complet et donc c'est un espace de Banach.

L'espace Euclidien \mathbb{R}^n

Soit \mathbb{R}^n l'espace vectoriel réel de dimension $n \in \mathbb{N}^*$, on notera sa base canonique e_1, e_2, \dots, e_n . où e_i est un vecteur de \mathbb{R}^n donné par:

$$(e_i)_j = \begin{cases} 0, & \text{si } i \neq j \\ 1, & \text{si } i = j \end{cases}, i, j = 1, 2, \dots, n$$

- **Produit scalaire:**

Quand on munit l'espace vectoriel \mathbb{R}^n par un produit scalaire alors si $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ et $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ sont deux vecteurs de \mathbb{R}^n , on aura:

$$\langle x, y \rangle = x^t y = \sum_{i=1}^n x_i y_i$$

On a les propriétés suivantes:

- ✓ $\langle x, y \rangle = \langle y, x \rangle$
- ✓ $\langle x, y + z \rangle = \langle x, y \rangle + \langle x, z \rangle$
- ✓ $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle$
- ✓ $\langle x, x \rangle \geq 0$ avec $\langle x, x \rangle = 0 \Leftrightarrow x = 0$

Le produit scalaire vérifie l'inégalité de Cauchy-Schwarz $\langle x, y \rangle^2 \leq \|x\|^2 \|y\|^2$ avec égalité si et seulement si x et y sont colinéaires.

L'inégalité de Cauchy Schwarz permet aussi de définir la mesure de l'angle géométrique θ entre deux vecteurs : comme $\langle x, y \rangle \leq \|x\| \|y\|$, si aucun des deux vecteurs n'est nul, alors le quotient $\frac{\langle x, y \rangle}{\|x\| \|y\|} = \cos\theta$, et le produit scalaire est nul lorsque les deux vecteurs sont orthogonaux.

Norme Euclidienne induite: Lorsque \mathbb{R}^n est munit de produit scalaire celui-ci induit ce qu'on appelle la norme Euclidienne:

$$\|x\| = \langle x, x \rangle^{1/2} = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}$$

Et la distance entre deux vecteurs de \mathbb{R}^n est donnée par:

$$d(x, y) = \|x - y\| = (\langle x - y, x - y \rangle)^{1/2}$$

On rappelle que sur \mathbb{R}^n toutes les normes sont équivalentes. On peut donc munir \mathbb{R}^n de différentes normes, les plus courantes sont les normes:

* **La norme p d'un vecteur:**

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

La norme euclidienne correspond à $p = 2$: c'est la seule des normes p qui soit issue d'un produit scalaire.

* **La norme infinie d'un vecteur** est définie de la manière suivante:

$$\|x\|_\infty = \max_i |x_i|$$

Notes:

- *Puisque toutes les normes sont équivalentes sur \mathbb{R}^n , si une suite est convergente pour une norme elle converge pour toutes les autres normes de \mathbb{R}^n .*
- *Un espace euclidien est un espace vectoriel normé dans lequel la norme est définie à partir d'un produit scalaire, Les espaces vectoriels normés complets et munis de produit scalaire constituent les espaces de Hilbert. Il s'en suit qu'un espace de Hilbert est caractérisé par la notion d'angle de deux vecteurs (c'est-à-dire de deux éléments de l'espace, ou plus précisément des vecteurs orientés qui joignent l'origine à ces deux éléments).*

Généralités sur les matrices

Étant donnée une matrice $(m \times n)$ $A = [a_{i,j}]_{i=1, \dots, m; j=1, \dots, n}$ à coefficients réels ou complexes, on notera $A = (a_{i,j})$. La transposée et conjuguée hermitienne ou transposée conjuguée de A sont des matrices $(n \times m)$ définies respectivement par:

$$[A^t]_{i,j} = [A]_{j,i} = (a_{j,i})$$

$$[A^H]_{i,j} = [A]_{j,i}^* = (a_{j,i}^*)$$

Où $a_{j,i}^*$ représente le transposé conjugué de l'élément $a_{j,i}$.

Pour une matrice carrée A , de taille n , rappelons maintenant la définition de quelques matrices particulières importantes.

Matrice symétrique réelle: $A^t = A$

Matrice hermitienne complexe: $A^H = A$

Matrice orthogonale réelle: $A^t A = A A^t = I$

Matrice unitaire complexe: $A^H A = A A^H = I$

Matrice normale: $A^H A = A A^H$

Propriété : (A, B deux matrices)

- $(A + B)^t = A^t + B^t$
- $(AB)^t = B^t A^t$
- $\begin{pmatrix} A & B \\ C & D \end{pmatrix}^t = \begin{pmatrix} A^t & C^t \\ B^t & D^t \end{pmatrix}$

Notations:

On note $M_{p,n}(\mathbb{R})$ l'espace vectoriel des matrices $p \times n$ à coefficient réel.

On note $M_n(\mathbb{R})$ l'algèbre des matrices carrées ($n \times n$) à coefficient réel.

Pour $A \in M_n(\mathbb{R})$ on note $\det(A)$ son déterminant et $\text{tr}(A)$ sa trace.

Rang d'une matrice:

Le rang d'une matrice correspond au nombre maximum de lignes ou de colonnes linéairement indépendantes. C'est aussi l'ordre du plus grand déterminant non nul. Soit k cet ordre, on dira que la matrice est de rang k .

Une matrice $A \in \mathbb{R}^{m \times n}$ est dite de rang plein si $\text{rang}(A) = \min(n, m)$ et on a:

$$\text{Rang}(A) = \text{rang}(A^t) = \text{rang}(AA^t) = \text{rang}(A^t A)$$

Noyau d'une matrice

Pour un système d'équations homogène $Ax=0$, $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$; l'ensemble de solutions x de ce système constitue un espace vectoriel appelé noyau de A , noté $\text{Ker}(A)$. La dimension de cet espace est notée N_A .

$$\text{Ker}(A) = \{x \in \mathbb{R}^n \mid Ax = 0\}$$

$$\text{Im}(A) = \{Ax \in \mathbb{R}^m, x \in \mathbb{R}^n\}$$

On a: $\text{Rang}(A) + N_A = n$

Notons qu'en anglais $\text{rang}(A)$ s'exprime par $\text{rank}(A)$ tandis que $\text{Im}(A)$ s'écrit $\text{range}(A)$ et $\text{Ker}(A)$ s'exprime par $\text{Null}(A)$.

Norme matricielle:

Une norme matricielle est une application $\|\cdot\| : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ telle que :

- $\forall A \in \mathbb{R}^{m \times n}$ alors $\|A\| > 0$ et $\|A\| = 0 \Leftrightarrow A = 0$
- $\forall A \in \mathbb{R}^{m \times n}, \forall B \in \mathbb{R}^{m \times n}$ alors $\|A + B\| \leq \|A\| + \|B\|$
- $\forall A \in \mathbb{R}^{m \times n}, \forall \alpha \in \mathbb{R}$, alors $\|\alpha A\| = |\alpha| \|A\|$

Ils existent plusieurs définitions pour les normes matricielles. Celles qui sont les plus utilisées, encore une fois nommées naturelles, découlent des normes vectorielles. Nous pouvons mieux caractériser les normes matricielles en introduisant les notions de norme compatible et de norme subordonnée à une norme vectorielle.

Définition:

Pour une matrice A , on dit qu'une norme matricielle $\|\cdot\|_M$ est compatible ou consistante avec une norme vectorielle $\|\cdot\|_V$ si

$$\|Ax\|_V \leq \|A\|_M \|x\|_V, \quad \forall x \in \mathbb{R}^n$$

D dorénavant on omettra les indices M et V .

Définition:

On dit qu'une norme matricielle $\| \cdot \|$ est sous-multiplicative si $\forall A \in R^{n \times m}$ et $\forall B \in R^{m \times q}$

$$\|AB\| \leq \|A\| \|B\|$$

Cette propriété n'est pas satisfaite par toutes les normes matricielles.

En faite à chaque norme vectorielle p , on peut lui faire correspondre une norme matricielle p de la façon suivante:

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p=1} \|Ax\|_p$$

Sous cette forme, il est difficile de calculer la norme d'une matrice, Nous allons énoncer quelques résultats qui vont rendre la tâche plus facile. Alors si $A = (a_{i,j}) \in Mn(R)$, on a:

La 1-norme

$$\|A\|_1 = \max_{j=1, \dots, n} \sum_{i=1}^m |a_{i,j}|$$

La norme 2 ou la norme euclidienne

$$\|A\|_2 = \max_i (\sigma_i(A))$$

La norme infinie

$$\|A\|_\infty = \max_i \sum_{j=1}^n |a_{i,j}|$$

La norme de Frobenius

$$\|A\|_F = \sqrt{\sum_{i=1}^n (A^t A)_{ii}} = \sqrt{\text{trace}(A^t A)}$$

Il est évident que $\|A\|_1 = \|A^t\|_\infty$, et si A est symétrique réelle on aura : $\|A\|_1 = \|A\|_\infty$

Trace d'une matrice:

La trace d'une matrice $A \in R^{n \times n}$, est égale à la somme des éléments de sa diagonale:

$$\text{trace}(A) = \sum_{i=1}^n a_{ii}$$

Avec

- $\text{trace}(AB) = \text{trace}(BA)$
- $\frac{\partial(\text{trace}(AB^t))}{\partial A} = B$
- $\frac{\partial(\text{trace}(ABA^t))}{\partial A} = 2AB$

Définition:

Soit A une matrice carrée d'ordre n . On dit que $\lambda \neq 0$ est valeur propre de A , si on peut trouver $x_\lambda \in R^n$, $x_\lambda \neq 0$ tel que $Ax_\lambda = \lambda x_\lambda$. Le vecteur x_λ est appelé vecteur propre associé à la valeur propre λ . L'ensemble des valeurs propres de A est le spectre de A .

Soit $A \in Mn(K)$ et $\lambda \in K$, alors

λ valeur propre de $A \Leftrightarrow (A - \lambda I_n)$ n'est pas inversible $\Leftrightarrow \det(A - \lambda I_n) = 0$

Soit $A \in Mn(R)$ et λ_i ses valeurs propres, alors

$$\det(A) = \prod_{i=1}^n \lambda_i$$

$$\text{Trace}(A) = \sum_{i=1}^n \lambda_i$$

Définition:

Soit $A \in Mn(K)$ ($K = R$ ou $K = C$), Notons λ_i les valeurs propres de A^*A , alors les valeurs singulières de A sont les $\sqrt{\lambda_i}$ $i = 1, \dots, n$. La matrice A^* désigne le conjugué transposé de A .

Propriétés générales:

- Les valeurs singulières sont des nombres réels positifs.
- Les valeurs singulières non nulles de A sont identiques à celles de A^* (invariance par l'opération transposé/conjugué)
- Les valeurs singulières non nulles sont au plus au nombre de $\min(n; m)$, la plus petite dimension de A .
- Les n valeurs singulière de λI_n , où $\lambda \in R$ sont toutes égales à λ .
- Les valeurs propres d'une matrice diagonale réelle sont égales aux valeurs absolues des éléments diagonaux ; pour une matrice de complexes, les valeurs propres sont égales aux modules des éléments de la diagonale.

Définition:

Le rayon spectral de $A \in Mn(K)$ ($K = R$ ou $K = C$), noté $\rho(A)$ est défini par:

$$\rho(A) = \max_i |\lambda_i|$$

Où $\lambda_i \in C$ (ou R) sont les valeurs propres de A .

Théorème:

Soit $\sigma_1(A)$ la plus grande valeur singulière de A . Alors

$$\|A\|_2 = \sqrt{\rho(A^*A)} = \sqrt{\rho(AA^*)} = \sigma_1(A)$$

En particulier, si A est hermitienne (ou symétrique réelle), alors

$$\|A\|_2 = \rho(A)$$

Puisque toute valeur propre de A est inférieure ou égale à $\|A\|$, alors

$$\rho(A) \leq \|A\|$$

On a:

$$\rho(A) = \lim_{k \rightarrow \infty} \|A^k\|^{\frac{1}{k}}$$

Par suite:

$$\rho(A) < 1 \text{ si et seulement si } A^k \rightarrow 0 \text{ quand } k \rightarrow \infty$$

Déterminant :

On appelle déterminant de A , le scalaire noté $\det(A)$ ou encore $|A|$ donné par l'une ou l'autre des expressions ci-dessous:

$$\det(A) = \sum_{k=1}^n (-1)^{i+k} a_{ik} \Delta_{ik}, \quad i = 1, \dots, n$$

$$\det(A) = \sum_{i=1}^n (-1)^{i+k} a_{ik} \Delta_{ik}, \quad n = 1, \dots, n$$

où Δ_{ik} est le déterminant de la matrice de dimension $((n - 1) \times (n - 1))$ obtenue en supprimant la i ème ligne et la k ème colonne de A :

Les formules précédentes indiquent que le déterminant de la matrice est indépendant de la ligne ou de la colonne choisie pour le développement. La quantité $(-1)^{i+k} \Delta_{ik}$ est appelée cofacteur de a_{ik} et les Δ_{ij} les mineurs principaux de A .

On aura une matrice de cofacteurs $C = (c_{i,j})$, avec $c_{i,j} = (-1)^{i+j} \Delta_{ij}$

Propriété du déterminant: (A,B deux matrices)

- $\det(AB) = \det(A) \det(B), \quad A, B \in \mathbb{R}^{n \times n}$
- $\det(A^{-1}) = \frac{1}{\det(A)}, \quad A \in \mathbb{R}^{n \times n}$
- $\det \begin{pmatrix} D & E \\ F & G \end{pmatrix} = \det(G) \det(D - EG^{-1}F) = \det(D) \det(G - FD^{-1}E), \quad D, G \in \mathbb{R}^{n \times n}$

Quelques théorèmes:

1. Si tous les éléments d'une ligne (colonne) d'une matrice A sont nuls, alors $\det(A) = 0$
2. Si tous les éléments d'une ligne (colonne) d'une matrice A sont multipliés par λ , alors $\det(A)$ est multiplié par λ .
3. Si B obtenu à partir de A en échangeant deux lignes (colonnes), alors $\det(B) = -\det(A)$
4. Si B obtenu à partir de A en faisant passer la ième ligne (colonne) par-dessus k lignes (colonnes), alors $\det(B) = (-1)^k \det(A)$
5. Si deux lignes (colonnes) d'une matrice sont identiques, alors $\det(A) = 0$
6. Si aux éléments d'une ligne (colonne) d'une matrice A on ajoute m fois les éléments correspondants d'une autre ligne (colonne), $\det(A)$ reste inchangé.

Matrice adjointe:

On appelle matrice adjointe, la matrice

$$\text{Adj}(A) = C^t, \quad \text{avec } C \text{ matrice des cofacteurs}$$

$$C = (c_{i,j}), \quad \text{avec } c_{i,j} = (-1)^{i+j} \Delta_{i,j}, \quad \Delta_{i,j} \text{ sont les mineurs principaux}$$

Matrice inverse:

Deux matrices A et B sont inverse l'une de l'autre, si leur produit est égal à la matrice unité.

$AB=I$, alors $B=A^{-1}$.

$$A^{-1} = \frac{\text{Adj}(A)}{\det(A)}$$

L'inverse généralisé ou pseudo-inverse utilisé dans la résolution des systèmes est noté: A^+ et satisfait les propriétés suivantes:

- $AA^+A=A$
- $A^+AA^+=A^+$
- $(A^+A)^t=A^+A$
- $(AA^+)^t=AA^+$

Comme déjà vue, le pseudo-inverse à gauche exprime la solution de systèmes d'équations sur déterminé ($Ax=b$). Alors que, le pseudo-inverse à droite exprime la solution de systèmes d'équations sous déterminé.

En faite, ces deux pseudo-inverses qu'on appelle généralement inverse généralisé de Moore-Penrose sont données par:

$$A^+ = \lim_{\delta \rightarrow 0} (AA^t + \delta^2 I)^{-1} A^t = \lim_{\delta \rightarrow 0} A^t (AA^t + \delta^2 I)^{-1}$$

Si les lignes de A sont linéairement indépendantes, on prendra $\delta = 0$.

Quelques propriété : ($A \in \mathbb{R}^{n \times m}, B \in \mathbb{R}^{m \times n}$ et si les inverses ci-dessous existent)

- $(A^{-1})^{-1} = A$
- $(A^t)^{-1} = (A^{-1})^t$

- $(AB)^{-1} = B^{-1}A^{-1}$
- $A(I_n + A)^{-1} = (I_n + A)^{-1}A$
- $A(I_n + A)^{-1} + (I_n + A)^{-1} = I_n$
- $A(I_n + BA)^{-1} = (I_m + AB)^{-1}A$
- $(I_m + AB)^{-1}A + A(I_n + BA)^{-1}B = I_m$

Théorème:

Soit une norme matricielle induite, I_d la matrice identité de $M_n(\mathbb{R})$ et $A \in M_n(\mathbb{R})$ telle que $\|A\| < 1$, alors la matrice $(I_d + A)$ est inversible et

$$\|(I_d + A)^{-1}\| \leq \frac{1}{1 - \|A\|}$$

Si la matrice $(I_d + A) \in M_n(\mathbb{R})$ est singulière, alors $\|A\| > 1$ pour toute norme matricielle.

Matrice définie positive/négative

Une matrice $A \in M_n(\mathbb{R})$ est définie positive, notée $A > 0 \Leftrightarrow \forall x \in \mathbb{R}^n, x^tAx > 0$

Une matrice $A \in M_n(\mathbb{R})$ est semi-définie positive, notée $A \geq 0 \Leftrightarrow \forall x \in \mathbb{R}^n, x^tAx \geq 0$

On a ainsi, une matrice $A \in M_n(\mathbb{R})$ est définie positive, si A est semi-définie positive et

$$x^tAx = 0 \rightarrow x = 0$$

On définit de façon analogue une matrice carrée semi-définie négative et définie négative.

Cela peut être vérifié à partir des valeurs propres qui peuvent être obtenues par la résolution du polynôme caractéristique:

$$A > 0 \Leftrightarrow \text{tous les } \lambda_i > 0$$

$$A \geq 0 \Leftrightarrow \text{tous les } \lambda_i \geq 0$$

$$A < 0 \Leftrightarrow \text{tous les } \lambda_i < 0$$

$$A \leq 0 \Leftrightarrow \text{tous les } \lambda_i \leq 0$$

On peut aussi utiliser le critère de Sylvester qui dit qu'une matrice est définie positive si tous ses sous-déterminant $\det(A_k) > 0$ avec $A_k = (a_{ij}), i, j = 1, \dots, k$. Les sous-déterminants sont parfois aussi appelés les mineurs principaux diagonaux.

Mineurs principaux d'une matrice

Soit A une matrice carrée symétrique de dimension (n, n) . Un mineur principal d'ordre k est le déterminant de la sous-matrice de A d'ordre k obtenue en supprimant $n - k$ lignes et les $n - k$ colonnes correspondantes dans A .

Exemple:

$$\text{Soit } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Les trois mineurs principaux d'ordre 1 de A sont: a_{11}, a_{22}, a_{33}

Les trois mineurs principaux d'ordre 2 de A sont : $\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}, \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}$ et $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$

Le mineur principal d'ordre 3 de A est : $\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$

Sous-déterminant $\det(A_k)$ ou mineurs principaux diagonaux

Soit A une matrice carrée symétrique de dimension (n, n) . Le mineur principal diagonal d'ordre k (noté Δ_k) de la matrice A est le déterminant de la matrice de taille (k, k) obtenue en éliminant les $n-k$ dernières lignes et $n-k$ dernières colonnes de la matrice A . Une matrice carrée d'ordre n admet n mineurs principaux diagonaux.

Exemple:

$$\text{Soit } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Le mineur principal d'ordre 1 de A est: a_{11}

Le mineur principal d'ordre 2 de A est : $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$

Le mineur principal d'ordre 3 de A est : $\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$

Caractérisation : Soit A une matrice carrée symétrique d'ordre n .

- A est définie positive \Leftrightarrow ses n mineurs principaux diagonaux Δ_k sont > 0 .
- A est semi-définie positive \Leftrightarrow tous ses mineurs principaux (et pas seulement diagonaux !) Δ_k sont ≥ 0 .
- A est définie négative \Leftrightarrow ses n mineurs principaux diagonaux Δ_k sont alternativement (< 0 Pour k impair) et (> 0 pour k pair).
- A est semi-définie négative \Leftrightarrow tous ses mineurs principaux Δ_k (et pas seulement diagonaux !) sont alternativement (≤ 0 Pour k impair) et (≥ 0 pour k pair).

De plus

- Si A est définie positive alors $\forall i = 1, \dots, n, a_{ii} > 0$
- Si A est semi-définie positive alors $\forall i = 1, \dots, n, a_{ii} \geq 0$

Nombre condition associé à une matrice:

On appelle nombre condition d'une matrice A associé à ses valeurs propres la quantité:

$$\kappa_\lambda(A) = \frac{\max_i |\lambda_i|}{\min_j |\lambda_j|} = \|A\|_p \|A^{-1}\|_p$$

On appelle les valeurs singulière d'une matrice A , les valeurs:

$$\sigma_i(A) = \sqrt{\lambda_i(A^*A)}$$

On démontre que toute matrice peut se décomposer sous la forme $A = U\Sigma V^t$; où U et V sont des matrices unitaires et Σ une matrice diagonale contenant les $\sigma_i(A)$ par ordre décroissant.

On appelle nombre condition de A la quantité:

$$\kappa(A) = \frac{\bar{\sigma}(A)}{\underline{\sigma}(A)}$$

Avec $\bar{\sigma}(A) = \max_i \sigma_i(A) = \sigma_1$ et $\underline{\sigma}(A) = \min_i \sigma_i(A) = \sigma_r$ où r est $\text{rang}(A)$.

On remarque que $\kappa(A) \geq 1$, s'il est grand, on dit que la matrice A est mal conditionnée. Les matrices mal conditionnées fournissent généralement des résultats de calcul peu stables.

Rappel de Calcul différentiel

$f: E \rightarrow F$

En général, nous considérons les fonctions $f: E \rightarrow F$; où E et F sont des espaces vectoriels de dimensions finies. Plus précisément, nous considérons le cas $E = \mathbb{R}^n$ et $F = \mathbb{R}^m$.

- Lorsque $m = n = 1$, une telle fonction est appelée fonction d'une variable réelle (numérique) à valeurs réelles.
- Lorsque $n = 1$ et $m > 1$, cette fonction est appelée une fonction vectorielle d'une variable réelle à valeurs réelles.
- Lorsque $n > 1$ et $m = 1$, la fonction est appelée fonction à valeurs réelles d'une variable vectorielle réelle, ou plus brièvement, un champ scalaire.
- Lorsque $n > 1$ et $m > 1$, elle est appelée fonction à valeurs vectorielles réelles d'une variable vectorielle, ou tout simplement champ de vecteurs (réel).

Nous avons préféré ne rien faire sur la notion de limite dans ce rappel pour nous concentrer exclusivement sur la notion de dérivée (dérivabilité) et de la différentielle (différentiabilité).

Nous allons voir plusieurs notions de dérivée d'un champ scalaire : dérivée par rapport à un vecteur, directionnelle, partielle, de Gâteaux et totale. Nous verrons que ces notions sont distinctes.

Fonctions numériques d'une variable réelle

Soit I un intervalle ouvert de \mathbb{R} et $f: I \rightarrow \mathbb{R}$ une application.

Si f est dérivable sur I , alors la dérivée de f au point a est donnée par:

$$f'(a) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}$$

Définition:

On dit que f est dérivable sur I si et seulement si $\forall x \in I, \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}$ existe et on note cette limite f'

Définition:

On dit que f est de classe C^1 sur I et on l'a noté $f \in C^1(I, \mathbb{R})$, si et seulement si est dérivable sur I et $f'(x)$ est continue sur I .

Exemple:

Soit $f: \mathbb{R} \rightarrow \mathbb{R}$ définie par:

$$f(x) = e^{2x} - x^2$$

Cette fonction est de classe C^1 sur \mathbb{R} car $\forall x \in \mathbb{R}, f'(x) = 2e^{2x} - 2x$ est continue sur \mathbb{R} .

Alors que pour la fonction définie $\mathbb{R}^+ \rightarrow \mathbb{R}$ par $f(x) = x^2 \sin \frac{1}{x}$ pour $x > 0$ et $f(0) = 0$, f est dérivable sur \mathbb{R}^+ :

$$\forall x > 0, \quad f'(x) = 2x \sin \frac{1}{x} - \cos \frac{1}{x^2}$$

f n'est pas de classe C^1 sur \mathbb{R}^+ car $f'(x)$ n'est pas continue en 0.

Remarque:

si f est définie de: $S \subset \mathbb{R}^p \rightarrow \mathbb{R}^q$, alors l'expression de la dérivée précédente n'a pas de sens parce que l'on ne peut pas diviser par un vecteur.

Par contre, on verra que si on fixe toutes les composantes du vecteur x sauf une, on peut définir les dérivées partielles de la fonction f .

Fonctions vectorielles d'une variable réelle

Soit I un intervalle de \mathbb{R} et une application $f: I \rightarrow \mathbb{R}^p$. Pour tout $x \in I$, on note $f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_p(x) \end{pmatrix}$.

Définition:

On dit que f est de classe C^1 sur I et on l'a note $f \in C^1(I, \mathbb{R}^p)$, si et seulement si pour tout $i=1,2,\dots,p$ $f_i(x)$ est de classe C^1 sur I et on écrit pour tout $x \in I$:

$$f'(x) = \begin{pmatrix} f_1'(x) \\ \vdots \\ f_p'(x) \end{pmatrix}$$

Exemple:

Soit $f: \mathbb{R}_+^* \rightarrow \mathbb{R}^2$ définie par:

$$\forall x \in \mathbb{R}_+^*, \quad f(x) = \begin{pmatrix} \sqrt{x} - x \\ e^x - \ln x \end{pmatrix}$$

On voit que f est de classe $C^1(\mathbb{R}_+^*)$ car $\forall x \in \mathbb{R}_+^*$, $f'(x) = \begin{pmatrix} \frac{1}{2\sqrt{x}} - 1 \\ e^x - \frac{1}{x} \end{pmatrix}$ est continue sur \mathbb{R}_+^* .

Calcul de dérivées partielles:

La dérivation d'une fonction d'une variable peut être généralisée. Par exemple pour une fonction à deux variables, les dérivées partielles d'une fonction de deux variables x et y se calculent de la façon suivante :

- Dérivée partielle par rapport à x qu'on note $\frac{\partial f}{\partial x}$: on considère que y est constant et on dérive la fonction comme fonction d'une variable x .
- Dérivée partielle par rapport à y qu'on note $\frac{\partial f}{\partial y}$: on considère que x est constant et on dérive par rapport à y .

On note la différentielle de la fonction f par df , qui est donnée par:

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy$$

On peut facilement définir:

- $\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right)$: on dérive deux fois par rapport à x
- $\frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial y} \right)$: on dérive deux fois par rapport à y
- $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial y} \right)$: on dérive une fois par rapport à y , puis une fois par rapport à x
- $\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial x} \right)$: on dérive une fois par rapport à x , puis une fois par rapport à y

Fonctions de plusieurs variables

Les fonctions de plusieurs variables sont des fonctions de chacune de leurs variables. Si elles sont dérivables par rapport à chacune des variables comme fonctions d'une variable alors elles admettent des dérivées partielles. Le calcul des dérivées partielles se fait donc comme le calcul des dérivées des fonctions réelles de la variable réelles (les autres variables sont considérées comme des constantes). Mais en dimension supérieure, dire qu'une fonction est dérivable, n'est pas seulement dire qu'elle a des dérivées partielles. On dit qu'une fonction

f est dérivable ou différentiable en un point si elle a une bonne approximation linéaire (ou affine) en ce point. Mais dans ce cas, la différentiable n'est pas un nombre mais une matrice.

❖ **Champ scalaire:** (fonction à valeurs réelles d'une variable vectorielle réelle)

Soit U un ouvert de \mathbb{R}^n et une fonction $f: U \rightarrow \mathbb{R}$. essayons d'étudier la dérivabilité de ce champ scalaire en un point x_0 par rapport à une direction donnée. Autrement dit, nous voulons savoir comment varie ce champ lorsqu'on bouge de x_0 vers un point proche.

Pour fixer les idées, supposons que $f(x_0)$ représente la température en un point x_0 donné dans une salle chauffée dont une fenêtre est ouverte. Si nous nous rapprochons de la fenêtre, la température tend à décroître, si nous nous rapprochons du chauffage, la température augmente. En général, la manière dont le champ change dépend de la direction selon laquelle nous nous dirigeons à partir de x_0 .

Supposons que nous spécifions cette direction par un second vecteur v . Plus précisément, supposons que nous allions de x_0 vers $x_0 + v$ le long de la ligne joignant x_0 et $x_0 + v$. Chaque point de ce segment est alors de la forme $x_0 + hv$, où $h \in \mathbb{R}$. La distance de x_0 à $x_0 + hv$ est $\|hv\| = |h|\|v\|$. Supposons que h est suffisamment petit de façon que $(x_0 + hv) \in U$ et essayons d'étudier le comportement lorsque $h \rightarrow 0$ du taux de variation de f sur le segment de ligne joignant x_0 à $x_0 + hv$.

$$\frac{f(x_0 + hv) - f(x_0)}{h}$$

Définition:

Soit $f: U \rightarrow \mathbb{R}$ un champ scalaire donné, soient $x_0 \in U$ et $v \in \mathbb{R}^n$. La dérivée de f en x_0 dans la direction \vec{v} , notée $f'(x_0, v)$ est définie par :

$$\frac{\partial f}{\partial v}(x_0) = f'(x_0, v) = \lim_{h \rightarrow 0} \frac{f(x_0 + hv) - f(x_0)}{h}$$

Cette notion n'a d'intérêt que si $v \neq 0$.

Dans le cas particulier où v est un vecteur unitaire (c'est à dire $\|v\| = 1$), la distance entre x_0 et $x_0 + hv$ est $|h|$. Dans ce cas, la limite précédente représente le taux de variations de f par unité de distance le long du segment joignant x_0 à $x_0 + hv$; la dérivée $f'(x_0, v)$ est appelée dérivée directionnelle.

En pratique, nous n'utiliserons que des dérivés dans les directions de la base canonique de \mathbb{R}^n c.à.d. (e_1, \dots, e_n) .

En particulier, si $v = e_k$ (le k -ième vecteur unitaire des coordonnées), nous utilisons alors une notation spéciale pour désigner $f'(x_0, e_k) = D_{e_k} f(x_0)$: on note

$$\frac{\partial f}{\partial x_k}(x_0) = \frac{\partial f}{\partial e_k}(x_0) = f'(x_0, e_k) = D_{e_k} f(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + he_k) - f(x_0)}{h}$$

Et on a pour un x quelconque:

$$\frac{\partial f}{\partial x_k}(x) = \frac{\partial f}{\partial e_k}(x) = f'(x, e_k) = D_{e_k} f(x) = \lim_{h \rightarrow 0} \frac{f(x + he_k) - f(x)}{h}$$

Où $\frac{\partial f}{\partial x_k}(x)$ est la dérivée partielle de f en x par rapport à la variable x_k (qu'on note aussi f'_{x_k}).

Si les coordonnées de x dans la base (e_1, \dots, e_n) sont (x_1, \dots, x_n) , alors les coordonnées de $(x + he_k)$ sont $(x_1, \dots, x_{k-1}, x_k+h, x_{k+1}, \dots, x_n)$.

Et l'expression de notre dérivée partielle devient:

$$\frac{\partial f}{\partial x_k}(x) = f'(x, e_k) = \lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_{k-1}, x_k + h, x_{k+1}, \dots, x_n) - f(x_1, x_2, \dots, x_n)}{h}$$

c'est à dire : on gèle toutes les variables x_j , pour $j \neq k$, et on dérive par rapport à x_k .

Autrement dit, on se ramène à la dérivation d'une fonction d'une variable !!!

Remarque:

Pour une fonction $f: U \subset \mathbb{R}^n \rightarrow \mathbb{R}$, f est dite Gâteaux-dérivable en x_0 si et seulement si f est dérivable en x_0 par rapport à toutes les directions (e_1, \dots, e_n) de v , et que l'application: $v \rightarrow f'(x_0, v)$ est linéaire. Cette dernière application linéaire est alors appelée dérivée de Gâteaux de f en x_0 .

Définition :

Soit U un ouvert de \mathbb{R}^n et $f: U \rightarrow \mathbb{R}$ une fonction

- Si f admet une dérivée dans la direction v en tout point x de U , on dit alors que : « f admet une dérivée dans la direction v sur U »
- Si pour tout $k \in [1, n]$, f admet une dérivée dans la direction e_k sur U et si toutes les dérivées partielles $\frac{\partial f}{\partial x_k}(x)$ de f sont continues, on dit que : « f est de classe C^1 sur U ».

Théorème:

Soit U un ouvert de \mathbb{R}^n et $f: U \rightarrow \mathbb{R}$, une fonction de classe C^1 sur U c.à.d. que f admet des dérivées partielles $\frac{\partial f}{\partial x_k}(x_0)$ (en x_0 de U) dans toutes les directions e_k et telle que tout $k=1, \dots, n$ l'application $x \rightarrow \frac{\partial f}{\partial x_k}(x)$ est continue sur U , alors f est différentiable en x_0 et on a:

$$\forall x \in U, df_{x_0}(x) = \sum_{k=1}^n \frac{\partial f}{\partial x_k}(x_0) x_k = \sum_{k=1}^n f'(x_0, e_k) x_k$$

On a introduit donc une nouvelle notion de différentiabilité d'une fonction évidemment plus forte que celle de la dérivabilité de la fonction selon un vecteur.

En effet, lorsqu'une fonction f est différentiable en x_0 , alors pour tout vecteur v de \mathbb{R}^n , f est dérivable en x_0 dans la direction v et on a: $D_v f(x_0) = df_{x_0}(v)$; la réciproque n'est pas vraie.

Définition :

Soit U un ouvert de \mathbb{R}^n et $f: U \rightarrow \mathbb{R}$, une fonction, f est de classe C^2 sur U si et seulement si f est de classe C^1 sur U , la fonction $\frac{\partial f}{\partial x_k}: U \rightarrow \mathbb{R}$ est différentiable et la fonction dérivée partielle seconde $\frac{\partial(\frac{\partial f}{\partial x_k})}{\partial x_k} = \frac{\partial^2 f}{\partial x_k^2}: U \rightarrow \mathbb{R}$ est continue sur U .

❖ **Champs vectoriels:** (fonction à valeurs vectorielles réelle d'une variable vectorielle réelle)

La théorie de la dérivation pour les champs de vecteurs est une extension directe de celle pour les champs scalaires.

Soit U un ouvert de \mathbb{R}^n et $f: U \rightarrow \mathbb{R}^m$ un champ vecteur donné, soient $x_0 \in U$ et $v \in \mathbb{R}^n$. On définit la dérivée totale de f en x_0 par $f'(x_0, v)$:

$$f'(x_0, v) = \lim_{h \rightarrow 0} \frac{f(x_0 + hv) - f(x_0)}{h}$$

Dès que cette limite existe. La dérivée $f'(x_0, v)$ est un vecteur de \mathbb{R}^m .

En effet, si $f = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} \in \mathbb{R}^m$ avec f_1, \dots, f_m sont des fonctions $U \rightarrow \mathbb{R}$, alors $\forall x_0 \in U, v \in \mathbb{R}^n$, on a:

$$f'(x_0, v) = \frac{\partial f}{\partial v}(x_0) = \begin{pmatrix} \frac{\partial f_1}{\partial v}(x_0) \\ \frac{\partial f_2}{\partial v}(x_0) \\ \vdots \\ \frac{\partial f_m}{\partial v}(x_0) \end{pmatrix} = \begin{pmatrix} f_1'(x_0, v) \\ f_2'(x_0, v) \\ \vdots \\ f_m'(x_0, v) \end{pmatrix}$$

Avec $\frac{\partial f_i}{\partial v}(x_0)$ déjà définie par: $\frac{\partial f_i}{\partial v}(x_0) = f_i'(x_0, v) = \sum_{k=1}^n f_i'(x_0, e_k) x_k$

Définition : (Jacobéenne)

Soit U un ouvert de \mathbb{R}^n et $f: U \rightarrow \mathbb{R}^m$ un champ vecteur donné, on note (quand elle existe) par $J_f(x) = Jf(x)$ la matrice Jacobéenne de f en x , c'est une matrice $m \times n$ définie par:

$$(J_f(x))_{i,j} = (Jf(x))_{i,j} = \frac{\partial f_i}{\partial x_j}(x) \in \mathbb{R} \text{ avec } i = 1, \dots, m \text{ et } j = 1, \dots, n$$

NB

On appelle aussi cette matrice la différentielle de f en x – on la note alors $df(x)$ ou $f'(x)$.

Ainsi une façon très simple pour vérifier si une fonction $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ est différentiable en un point x_0 ; il suffit de calculer sa matrice Jacobéenne donc tous ses dérivée partielles si il y a un certain i_0 pour lequel $\frac{\partial f_{i_0}}{\partial x_j}(x_0)$ pour un j n'existe pas donc f n'est pas différentiable.

Composition de Fonctions:

Soit $f: \mathbb{R}^p \rightarrow \mathbb{R}^n$ une fonction différentiable en $x \in \mathbb{R}^p$ et $g: \mathbb{R}^n \rightarrow \mathbb{R}^q$ une fonction différentiable en $y = f(x) \in \mathbb{R}^n$, alors la composée $g \circ f$ est différentiable en x :

$$d(g \circ f)_x = d_{g(f(x))}(df_x(h))$$

Et on a:

$$d(g \circ f)_x = J(g \circ f)_x = J_g(f(x)) \cdot J_f(x)$$

Ce résultat peut s'avérer très utile lorsqu'on veut exprimer la différentielle de fonctions un peu "compliquée".

Gradient:

Le gradient de f en x est défini comme la transposée de la matrice Jacobéenne de f en x donc c'est un matrice $n \times m$:

$$\nabla f(x) = (J_f(x))^t$$

Remarque importante : Dans le cas particulier $m = 1$ donc ($f: U \rightarrow \mathbb{R}$), on dira que $\nabla f(x)$ est le vecteur colonne de \mathbb{R}^n .

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix}$$

Remarquons que $f(x)$ est un nombre alors que $\nabla f(x)$ est un vecteur.

De plus pour $x \in U \subset \mathbb{R}^n$ et $h \in \mathbb{R}^n$, on a:

$$\frac{\partial f}{\partial h}(x) = (\nabla f(x))^t h$$

Toujours, pour le cas particulier $m = 1$ donc ($f: U \rightarrow \mathbb{R}$), on note (quand elle existe) par $\nabla^2 f(x)$ la matrice carrée $n \times n$ définie par:

$$(\nabla^2 f(x))_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}, \quad \forall i, j = 1, \dots, n$$

La matrice $\nabla^2 f(x)$ est appelée matrice Hessienne de f en x qu'on notera aussi $H_f(x)$.

Sachant que d'après Schwarz et si f est de classe C^2 , on a; $\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$.

On peut aussi facilement vérifier que la matrice Hessienne de f est le Jacobien du gradient de f ou le gradient de la Jacobienne de f

$$\nabla^2 f(x) = J_{\nabla f}(x) = \nabla J_f(x)$$

Remarque importante:

Il existe une autre caractérisation de la dérivée et de la différentielle de fonctions, cette caractérisation est en relation directe avec l'approximation de fonction aux points de voisinage et les développements limités.

Théorème: (dérivée)

Une fonction f de $\mathbb{R} \rightarrow \mathbb{R}$ est dérivable en un point x_0 de nombre de dérivée $f'(x_0)$, si et seulement si, il existe une fonction ε telle que pour tout h avec $(x_0 + h)$ appartient au domaine de définition de f , on ait:

$$f(x_0 + h) = f(x_0) + hf'(x_0) + h\varepsilon(h); \text{ avec } \lim_{h \rightarrow 0} \varepsilon(h) = 0.$$

Théorème: (différentielle)

Pour une fonction $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, f est différentielle en $x_0 \in \mathbb{R}^n$ si pour tout $h \in \mathbb{R}^n$, il existe une application linéaire L de $\mathbb{R}^n \rightarrow \mathbb{R}^m$ telle que:

$$\lim_{\|h\| \rightarrow 0} \frac{f(x_0 + h) - f(x_0) - L(h)}{\|h\|} = 0$$

Ou en posant $x - x_0 = h$

$$\lim_{\|x\| \rightarrow \|x_0\|} \frac{f(x) - f(x_0) - L(x - x_0)}{\|x - x_0\|} = 0$$

L est donc la différentielle de f en x et se note: $df(x)$.

Cette application lorsqu'elle existe elle est unique, on peut aussi l'écrire sous forme suivante:

$$\|f(x_0 + h) - f(x_0) - L(h)\| = r(h), \quad \text{avec } r(h) = O(\|h\|)$$

Développements limités:

Le problème de base dans le calcul numérique est de pouvoir trouver une approximation de la valeur d'une fonction f en un point x_0 donné. Pour cela, on suppose que l'on dispose de la valeur de la fonction et de ses dérivées successives en un point x_1 . Par exemple, on cherche à calculer $\cos(0,1)$ connaissant $\cos(0) = 1$:

Le développement d'une fonction au premier ordre est une égalité locale. C'est-à dire que l'on peut considérer (définition de limite) que si l'on est suffisamment proche de x_0 , ($h \ll 1$) l'approximation : $f(x_0 + h) \approx f(x_0) + hf'(x_0)$ est bonne.

Théorème: (Série de Taylor)

Soit f une fonction de $\mathbb{R} \rightarrow \mathbb{R}$ de classe C^∞ , x_0 un point et x un point du voisinage de x_0 , alors on a:

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0)f'(x_0) + (x - x_0)^2 \frac{f''(x_0)}{2!} + (x - x_0)^3 \frac{f'''(x_0)}{3!} + \dots \dots \\ &= \sum_{k=0}^{\infty} (x - x_0)^k \frac{f^{(k)}(x_0)}{k!} \end{aligned}$$

Ce théorème signifie que la valeur de f en x se déduit de la valeur de f et de toutes ses dérivées au point x_0 . Appliquer la formule de Taylor sur calculateur pose un certain nombre de difficultés :

- Comment choisir x_0 ?
- Le calcul des dérivées $f^{(k)}(x_0)$ est-il facile ?
- A quel ordre peut-on arrêter la sommation ? Que vaut alors l'erreur commise sur la valeur calculée de $f(x)$?

Exemple:

$$f(x) = \cos(x)$$

$$f^{(k)}(x) = \cos\left(x + k\frac{\pi}{2}\right), \quad k=1, \dots, n$$

$$f(x) = \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \dots$$

Choisissons $x_0 = 0$ et $x = 0,1$ alors, on a:

$$f(0,1) = \cos(0,1) \approx 0.995, \quad k = 1$$

$$f(0,1) = \cos(0,1) \approx 0.99500416 \quad k = 2$$

Plus k est grand plus on s'approche mieux à la valeur exacte.

Théorème: (Théorème de Taylor)

Soit f une fonction de $\mathbb{R} \rightarrow \mathbb{R}$ de classe C^{n+1} sur un intervalle $[a, b]$, soient x et x_0 appartenant à $[a, b]$, alors il existe $\varepsilon(x)$ appartenant à un ouvert entre x et x_0 tel que:

$$f(x) = \underbrace{\sum_{k=0}^n (x - x_0)^k \frac{f^{(k)}(x_0)}{k!}}_{\text{approximant de } f} + \underbrace{(x - x_0)^{(n+1)} \frac{f^{(n+1)}(\varepsilon(x))}{(n+1)!}}_{\text{erreur}}$$

L'intérêt de ce théorème par rapport au précédent est qu'il permet d'obtenir un majorant de l'erreur connaissant un majorant de $f^{(n+1)}(x)$.

Exemple:

1) On peut facilement retrouver le résultat de l'exemple précédent.

2) $f(x) = e^x, |x| < \infty, f^{(k)}(x) = e^x, \forall k$ et prenons $x_0=0$

$$e^x = \sum_{k=0}^n \frac{x^k}{k!} + \underbrace{\frac{e^{\varepsilon(x)}}{(n+1)!} x^{n+1}}_{\text{erreur}}$$

L'erreur tend vers 0 quand n tend vers ∞ et on a:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \dots$$

Certaines fonctions peuvent poser des problèmes ; celles pour lesquelles, on ne peut garantir la convergence du terme d'erreur vers 0 sur l'intervalle de définition. C'est le cas de la fonction $\ln x$ ci-dessous.

1) $f(x) = \ln x, 0 < x \leq 2, f^{(k)}(x) = (-1)^{k+1}(k-1)!/x^k, \forall k \geq 1$ avec $x_0=0$

$$\ln x = \sum_{k=0}^n (-1)^{k+1} \frac{(x-1)^k}{k} + \underbrace{(-1)^n \frac{1}{(n+1)} \frac{(x-1)^{n+1}}{(\varepsilon(x))^{n+1}} x^{n+1}}_{\text{erreur}}$$

On ne peut garantir que l'erreur tend vers 0 quand n tend vers l'infini pour tout le domaine $0 < x \leq 2$ et donc le choix de x et x_0 est crucial pour ces approximations.

Théorème: (Théorème de Taylor: 2^{ème} forme)

Soit f une fonction de $\mathbb{R} \rightarrow \mathbb{R}$ de classe C^{n+1} sur un intervalle $[a, b]$, soient x et h tels que x et $(x+h)$ appartiennent à $[a, b]$. Alors il existe $\varepsilon(x)$ appartenant à un ouvert entre x et $(x+h)$ tel que:

$$f(x+h) = \underbrace{\sum_{k=0}^n \frac{f^{(k)}(x)}{k!} h^k}_{\text{approximant de } f} + \underbrace{\frac{f^{(n+1)}(\varepsilon(h))}{(n+1)!} h^{(n+1)}}_{\text{erreur}}$$

Cette deuxième forme du théorème de Taylor permet d'introduire la notion d'infiniment petit et de vitesse de convergence. On notera l'erreur E . Elle satisfait :

$$E_{n+1} = O(h^{(n+1)}) \Leftrightarrow |E_{n+1}| = C(|h|^{(n+1)})$$

Avec

$$C = \left| \max_{\varepsilon(h)} \frac{f^{(n+1)}(\varepsilon(h))}{(n+1)!} \right|$$

A partir de ces développements, on peut aussi extraire des formules d'approximation des dérivées pour le calcul numérique, par exemple pour une fonction $\mathbb{R} \rightarrow \mathbb{R}$:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \quad \text{et} \quad f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

L'extension de ces résultats aux fonctions multivariées à valeurs réelles est donnée par le théorème suivant:

Théorème: (Théorème de Taylor-Young: 2^{ème} forme pour des fonctions multivariées à valeurs réelles)

Soit f une fonction de $\mathbb{R}^n \rightarrow \mathbb{R}$ de classe C^2 , alors pour tout x de \mathbb{R}^n et h suffisamment petit,

$$f(x+h) = f(x) + \nabla f(x)^t h + \frac{1}{2} h^t \nabla^2 f(x) h + O(\|h\|^2)$$

Dans l'énoncé de ce théorème, la notation $O(\|h\|^2)$ pour k entier non nul signifie une expression qui tend vers 0 plus vite que $\|h\|^2$ (c'est à dire, si on la divise par $\|h\|^2$, le résultat tend vers 0 quand $\|h\|$ tend vers 0).

Cette formule de Taylor permet de disposer d'un modèle polynomial de degré 2 pour la fonction f . Il n'a de sens que localement autour du point x . Son intérêt est de permettre des calculs explicites que ne permet pas la fonction f en général.

Calcul numérique sur ordinateur

virgule flottante:

Sur un ordinateur, en calcul numérique, les réels sont représentés (approximés) par des nombres en virgule flottante. Généralement, on dispose de flottants en simple précision (codés avec 32 bits sur la machine) et des flottants en double précision (sur 64 bits).

La norme prévoit des codages pour des nombres flottants spéciaux (on dit ! anormaux "). Les plus importants sont plus ou moins l'infini et NaN (acronyme pour Not a Number). Les deux infinis apparaissent naturellement lorsqu'un flottant atteint une valeur qui dépasse, en valeur absolue, la plus grande valeur représentable avec le codage utilisé. Le flottant NaN apparaît par exemple, lorsqu'on calcule la racine carrée d'un nombre négatif ou dans une division par zéro.

Arithmétique flottante:

Définition:

On définit l'épsilon machine, noté $\varepsilon_{\text{machine}}$, comme la demi-distance entre le flottant 1 et le plus petit flottant supérieur ou égal à 1.

Epsilon machine:

En effet, il suffit de préciser un epsilon machine pour préciser une arithmétique flottante. L'idée, c'est qu'il est possible de spécifier les arithmétiques flottantes (arrondi sur les données, arrondi sur les opérations), en s'appuyant sur l'epsilon machine. On peut ainsi tenir des raisonnements sur les algorithmes numériques sans jamais se soucier de la précision avec laquelle on travaille.

Grand O d'epsilon machine:

En calcul numérique, on écrit souvent que le résultat d'un calcul est égal au résultat théorique, avec une erreur e vérifiant :

$$e = O(\epsilon_{\text{machine}})$$

Cela en imaginant que le calcul qui a donné l'erreur est effectuée sur des arithmétiques flottantes de plus en plus précises et donc pour lesquelles $\epsilon_{\text{machine}}$ tend vers zéro.

Signifiant:

$$\forall \epsilon_{\text{machine}}, \exists c > 0 \text{ telque } e \leq c\epsilon_{\text{machine}}$$

Erreurs et précision:

Pour évaluer la précision d'un résultat, on doit connaître parfaitement les erreurs qui ont été commises. Donnons trois exemples d'erreurs les plus importantes:

- Les erreurs d'arrondi sont imposées par le calculateur. La représentation d'un nombre en mémoire de l'ordinateur étant finie, tout nombre réel n'est connu qu'avec une précision donnée de n chiffres significatifs. Pour un nombre quelconque compris entre 0 et 1, la machine écrira par exemple
 $x = 0. a_1a_2a_3a_{..}a_n$
 Lors de la manipulation de ces nombres, la machine devra choisir entre la troncature ou l'arrondi à la décimale la plus proche. On comprend comment, à plus grande échelle, ces erreurs peuvent induire des problèmes de précision.
- Les erreurs de troncature sont liées à la précision de l'algorithme utilisé. Elles peuvent être contrôlées par l'algorithme lui-même. Si une fonction est approchée par son développement de Taylor, l'erreur de troncature sera obtenue par une évaluation du reste du développement. Son contrôle sera obtenu par une majoration de ce reste.
- Les erreurs de méthode se produisent lorsqu'une expression est mal équilibrée et mélange des valeurs dont la différence est importante. C'est un problème de calibration numérique qui est sensible aux erreurs d'arrondi. Dans la plupart des cas, l'algorithme doit être modifié.

Dans les processus récurrents ou itératifs, les erreurs s'ajoutent et se propagent d'une étape à l'autre. Ce qui a pour effet d'amplifier l'erreur globale et de diminuer la précision du calcul. La propagation des erreurs dans diverses parties du calcul a pour conséquence d'ajouter de l'imprécision des résultats de calcul.

Méthodes itératives

Le principe des méthodes itératives consiste à se rapprocher de la solution d'un problème en construisant une suite qui converge au mieux vers la solution (en arrêtant l'itération lorsque la différence entre deux pas successifs devient inférieure à une certaine valeur).

Par exemple en analyse numérique:

Dans les méthodes d'approximations successives, l'équation $f(x) = 0$; $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ est équivalente à l'équation $x = \varphi(x)$ (possédant les mêmes solutions) qui peut être remplacée par l'étude de la suite numérique convergente:

$$x_{n+1} = \varphi(x_n)$$

Cette suite nous permet d'obtenir la solution de l'équation en un nombre fini d'itérations, en général on prend $\varphi(x_n) = x - cf(x)$ avec c un nombre non nul ou une matrice bien choisie.

- Dans la méthode de Lagrange, on remplace la fonction f par le segment de droite passant par les points $(a, f(a))$ et $(b, f(b))$; on aura:

$$\varphi(x) = a - f(a) \frac{x - a}{f(x) - f(a)}$$

- Dans la méthode Newton, on remplace la fonction f entre les points d'abscisses a et b par la tangente à la courbe en ces points; on aura:

$$\varphi(x) = x - \frac{f(x)}{f'(x)}$$

- Pour la méthode de la Sécante, encore appelée méthode de la fausse position ou "regula-falsi", on approxime dans la méthode Newton la dérivée $f'(x)$ par:

$$f'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}, \text{ et on aura:}$$

$$x_{n+1} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$$

De même dans la résolution itérative des systèmes d'équations linéaires, la méthode consiste à représenter le système d'équations sous la forme d'une équation matricielle récurrente qui permet, à partir d'un vecteur initial fixé de construire une suite de vecteurs dont on espère qu'elle converge vers la solution du système. Plus précisément, pour le système linéaire d'équations $Ax = b$, si on décompose A sous la forme $A = M - N$, il apparaît que la solution x de $Ax = b$ est également solution de $Mx = Nx + b$. Remarquer que cette équation est une équation de la forme $x = f(x)$. En d'autres termes, x est un point fixe de l'équation de récurrence ci-dessous en initialisant l'algorithme par un vecteur arbitraire x_0 , on aura:

$$x_{n+1} = M^{-1}Nx_n + M^{-1}b$$

On démontre que la convergence de cette méthode ne dépend pas du choix de x_0 et la méthode itérative converge si et seulement si le rayon spectral de la matrice $M^{-1}N$ est strictement inférieur à 1, $\rho(M^{-1}N) < 1$. Selon les choix des matrices M et N on a différentes méthodes itératives. Il serait souhaitable que l'inversion de M soit simple, ce qui conduit souvent à choisir M égale à la partie diagonale ou à la partie triangulaire, par exemple inférieure, de A . Ces choix conduisent respectivement aux méthodes de Jacobi et de Gauss-Siedel.

Convergence et stabilité

Les méthodes numériques utilisées pour résoudre un problème approché conduisent à un résultat qui est toujours entaché d'erreur. Cette erreur doit être suffisamment petite pour que la solution numérique converge vers la solution réelle. Dans ce cas l'algorithme (ou la méthode) est dit convergent. Si un raisonnement mathématique permet de montrer qu'une méthode diverge, elle ne pourra en aucun cas être utilisée sur un calculateur. En revanche, si la méthode converge il se peut qu'en pratique elle diverge.

La vitesse de convergence est un facteur important de la qualité des algorithmes. Si la vitesse de convergence est élevée, l'algorithme converge rapidement et le temps de calcul est moindre. Ces préoccupations de rapidité de convergence ont conduit à diversifier les modes de convergence et à chercher des processus optimaux.

La stabilité de l'algorithme: elle garantit que les erreurs ne s'amplifient pas au cours du déroulement de l'algorithme et que la méthode reste stable.

Stabilité des solutions: elle intervient dans les problèmes énoncés et qui est bien mise en évidence par les techniques de perturbations. Lorsqu'un problème (P) admet une solution, il est intéressant d'envisager le problème perturbé, noté (P_ε) , où ε est un petit paramètre, et de se demander si les solutions du système perturbé sont voisines de la solution du système non perturbé. Il n'existe pas de théorie générale qui réponde à cette question.

Complexité algorithmique:

La complexité d'un algorithme s'exprime généralement en fonction du nombre d'opérations à réaliser pour obtenir la solution du problème à résoudre. Précisant, si ce paramètre est très important, dans certaines applications, la quantité de mémoire informatique nécessaire au traitement peut s'avérer cruciale pour le choix d'un algorithme.

En fait, les problèmes traités sur un ordinateur se répartissent en deux grandes catégories selon qu'une valeur numérique est attendue (problèmes de calcul) ou qu'une réponse par oui ou non est souhaitée (problème de décision).

Les propriétés des algorithmes ont été étudiées dans les années 1930 par le mathématicien Alan Turing (1912-1954) qui inventa la machine qui porte son nom. Turing, en démontrant que les problèmes qui ne pouvaient pas être résolus par sa machine symbolique n'avaient pas d'algorithme, fixa les limites de la calculabilité. Depuis alors, on classe les problèmes en deux grandes catégories : les problèmes pour lesquels il n'existe pas d'algorithme et les problèmes pour lesquels un algorithme existe. Parmi ces derniers, on mesure l'efficacité de l'algorithme selon la croissance de la durée de leur exécution en fonction de la taille du problème. Pour un problème de taille n , on considère comme efficaces les algorithmes dont la croissance est polynomiale et inefficaces ou difficilement exploitables les algorithmes dont la croissance est exponentielle. On dit qu'un problème n'est pas décidable s'il n'est pas soluble. On distingue plusieurs classes:

La classe P (Polynomial) représente la classe des langages décidables en un temps polynomial : ce sont les problèmes qui admettent une solution sur une machine de Turing en temps polynomial. La résolution d'un problème est obtenue en un temps inférieur à une puissance donnée de la taille n du problème : si la taille n du problème augmente, le nombre d'étapes de l'algorithme reste toujours plus petit qu'une certaine puissance de n .

La classe NP (Non deterministic polynomial) représente la classe des langages décidables en temps non déterministe polynomial. Ce sont des problèmes pour lesquels, si une solution est proposée, on peut vérifier que cette solution répond bien au problème en un temps polynomial. Pour certains problèmes de cette classe, on ne connaît aucun algorithme polynomial. On sait que la classe P est contenue dans la classe NP et on suppose que $NP = P$.

La classe NP-complet représente les problèmes de la classe NP qui sont liés : si un problème de cette classe peut être résolu par un algorithme en temps polynomial, alors tous les problèmes de la classe NP seront solubles par un algorithme efficace. Si on trouve un tel algorithme, on aura alors identité des classes P et NP.

La complexité d'un algorithme est définie comme le terme dominant de la formule qui exprime le nombre d'opérations à réaliser lorsque la dimension caractéristique (par exemple la taille de la matrice) du problème croît. Pour un problème de dimension n , on pourra définir cette complexité comme une grandeur $T(n)$ telle que:

$$\lim_{n \rightarrow \infty} \frac{\text{nombre d'opérations en dimension } n}{T(n)} = 1$$

Dans de nombreuses situations, on se contente d'un ordre de grandeur de la complexité algorithmique. Notons que souvent, on ne compte que le nombre de multiplications et de divisions, les additions et les soustractions n'étant pas prises en compte.

Problèmes bien posés, problèmes raides

Un problème (P) est mathématiquement bien posé si le problème (P) admet une solution unique qui est stable au sens de Hadamard, c'est-à-dire qui dépend continûment des données initiales. Un problème numérique est dit numériquement bien posé si la continuité de la solution est suffisamment bonne par rapport aux conditions initiales pour que la solution ne soit pas perturbée par une erreur initiale ou de petites erreurs d'arrondi.

Conditionnement:

Un autres aspect important lié à la résolution algorithmique réside dans la robustesse de la solution du problème obtenue vis à vis d'erreurs sur la connaissance précise des valeurs des paramètres du problème et sur la précision des calculs en machine.

Un problème sera dit bien conditionné lorsque sa solution variera peu lors d'une faible perturbation de ses paramètres. Considérons plus particulièrement le cas simple et qui nous intéresse ici les systèmes d'équations linéaires:

Conditionnement d'un système linéaire:

Considérons le système linéaire $Ax=b$ suivant:

$$\begin{pmatrix} 23 & 9 & 12 \\ 12 & 10 & 1 \\ 14 & -12 & 25 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 44 \\ 23 \\ 27 \end{pmatrix}, \text{ dont la solution est } x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Remarquons que la matrice A est inversible et admet trois valeurs propres $\lambda_1 \cong 36,16$, $\lambda_2 \cong 0,056$ et $\lambda_3 \cong 21,79$

Considérons le problème suivant dans lequel le vecteur b est légèrement perturbé

$$\begin{pmatrix} 23 & 9 & 12 \\ 12 & 10 & 1 \\ 14 & -12 & 25 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 44,44 \\ 23,77 \\ 27,73 \end{pmatrix}, \text{ dont la solution est } x = \begin{pmatrix} 6,23 \\ 4,73 \\ 4,69 \end{pmatrix}$$

Remarquons qu'une erreur de 1/100 sur les données entraîne une erreur relative de l'ordre de 5 sur la solution, les composantes du vecteur solution sont multipliées par 5. De même, si on perturbe les éléments de la matrice

$$\begin{pmatrix} 23,23 & 9,09 & 12,12 \\ 12,12 & 9,9 & 1,01 \\ 14,14 & -11,88 & 25,25 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 44 \\ 23 \\ 27 \end{pmatrix}, \text{ dont la solution est } x = \begin{pmatrix} 6,89 \\ 5,56 \\ 5,40 \end{pmatrix}$$

Une erreur de 1/100 sur les données provoque une erreur de l'ordre de 6. L'amplification des erreurs relatives est d'environ 600 et ça peut être beaucoup plus pour un autre système:

Considérons un autre système linéaire $Ax=b$

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}, \text{ dont la solution est } x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Alors que:

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32,1 \\ 22,9 \\ 33,1 \\ 30,9 \end{pmatrix}, \text{ a une solution } x = \begin{pmatrix} 9,2 \\ -12,6 \\ 4,5 \\ -1,1 \end{pmatrix} \text{ et}$$

$$\begin{pmatrix} 10 & 7 & 8,1 & 7,2 \\ 7,08 & 5,04 & 6 & 5 \\ 8 & 5,98 & 9,89 & 9 \\ 6,99 & 4,99 & 9 & 9,98 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}, \text{ a une solution } x = \begin{pmatrix} -81 \\ 137 \\ -34 \\ 22 \end{pmatrix}$$

Sans commentaires.....

De tels systèmes linéaires sont dites mal conditionnés. Il y a en effet beaucoup de définitions possibles du conditionnement, qui dépendent du problème à résoudre de manière générale on peut

dire qu'un problème est mal conditionné si de petites erreurs/perturbations dans les données provoquent de grands changements dans les résultats.

On peut essayer de quantifier le conditionnement d'un système linéaire. En envisageant le problème du point de vue algébrique. Soit A une matrice inversible et $Ax=b$ un système linéaire. Étudions sa perturbation en général:

$$(A + \delta A)(x + \delta x) = b + \delta b$$

Où δA et δb sont les perturbations sur A et b dues aux erreurs d'arrondi et δx l'erreur commise sur la résolution du système linéaire. Comme $Ax=b$, il vient

$$(A + \delta A)\delta x = \delta b - \delta A A^{-1} b$$

Si la matrice $(I + A^{-1}\delta A)$ est inversible, alors:

$$\delta x = (I + A^{-1}\delta A)^{-1} A^{-1} (\delta b - \delta A A^{-1} b)$$

D'où la majoration:

$$\|\delta x\| \leq \frac{\|A^{-1}\| \cdot (\|\delta b\| + \|\delta A\| \cdot \|A^{-1}b\|)}{1 - \|A^{-1}\| \|\delta A\|}$$

Comme $\|A^{-1}b\| = \|x\| \geq \frac{\|b\|}{\|A\|}$ et si $\text{cond}(A) = \kappa(A) = \|A\| \|A^{-1}\|$ vérifie $\kappa(A) \frac{\|\delta A\|}{\|A\|} < 1$, on aura:

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right)$$

Optimisation:

Comme l'objectif de ce document est de présenter une introduction aux méthodes d'optimisation, on retient que l'on cherche à réduire le nombre d'opérations et en premier lieu le nombre de multiplications. Pour un problème donné, on choisira alors toujours l'algorithme qui donne la meilleure précision sur les résultats et qui minimise le temps de calcul et l'espace mémoire nécessaire.

Chapitre 1

Introduction à l'optimisation

Introduction

Les mots "optimiser", optimisation reflète comme toute attitude naturelle dans la vie courante l'idée de faire le meilleur possible. En effet, nous sommes fréquemment confrontés à des problèmes "d'optimisation" plus ou moins complexes. Cela peut commencer au moment où l'on tente de ranger son bureau ou de faire ses courses, de placer son mobilier, et aller jusqu'à un processus industriel et planification de différentes tâches. En effet, l'être humain cherche toujours à améliorer sa vie quotidienne; sans se rendre compte, il cherche à vrai dire à optimiser en minimisant ses dépenses et en maximisant ses biens selon le budget qu'il a et les contraintes du mode de vie qu'il mène. Si tout le monde est d'accord qu'il vaut mieux d'être riche et en bonne santé que pauvre et malade, on se trouve rarement en pratique face à de choix aussi clair. Le plus souvent consiste à faire un compromis entre plusieurs critères en tête qui ne sont toujours pas comparables (voir contradictoires). C'est pourquoi beaucoup de choix sont difficiles, en effet optimiser n'a de sens que relativement par rapport aux contraintes considérées.

En général, l'optimisation permet de se trouver dans la meilleure situation pour obtenir un gain d'effort, de temps, d'argent, d'énergie, de matière première, ou encore de satisfaction dans des domaines d'applications extrêmement variés : optimisation d'un trajet, de la forme d'un objet, d'un prix de vente, d'une réaction chimique, du contrôle aérien, du fonctionnement d'un moteur, de la gestion des lignes ferroviaires, du choix des investissements économiques, de la construction d'un grand projet ou d'un navire, etc..

Dans le domaine technique, les choix possibles représentent souvent un ensemble nombrable (voir innombrable), et une telle énumération exhaustive des possibilités de choix est impensable. Il faut alors un "algorithme" plus performant, c'est à-dire une méthode pour aller vers "la" solution en explorant le moins de "moins bonnes" solutions possible. Par exemple en engineering, optimiser la conception d'une installation sous les contraintes de budget (horaires de travail, temps d'exécution, coût d'investissement,...), sécurité, fiabilité, performances et aussi l'intérêt général du projet, tous ces facteurs (contraintes) représentent un grand défi pour le concepteur. C'est pour cela que l'optimisation représente actuellement un outil précieux d'aide à la décision en politique économique.

Une fois bien défini l'échelle de mesure des performances, lorsque les contraintes sont explicitement spécifiées, il fera mieux de chercher la solution réputée "optimale. Néanmoins, on peut parler de "solution sous-optimale" lorsqu'on est prêt à admettre qu'il se pourrait qu'on puisse encore faire mieux, mais, sans précision sur le "combien mieux", ou "combien moins bon".

Si aujourd'hui, tous les systèmes susceptibles d'être décrits par un modèle mathématique sont optimisés. La qualité des résultats et des prédictions dépend de la pertinence du modèle, de l'efficacité de l'algorithme et des moyens pour le traitement numérique. De là provient l'intérêt des techniques mathématiques permettant de bien formuler ces notions sous forme de modèle de fonction de coût ou objectif et d'exprimer plus clairement les contraintes du problème. Et toute l'opération d'optimisation se réduit donc à la recherche d'extrémums (max ou min) de ces fonctions objectifs sous les contraintes imposées. On se trouve alors dans le cadre de la programmation mathématique "PM" qui fait intervenir plusieurs étapes à commencer par la modélisation qui permet de mettre en forme le problème étudié sous la forme mathématique appropriée à la mise en œuvre d'algorithmes; puis l'optimisation ainsi que l'analyse numérique pour la recherche d'extrémums.

De ce fait, l'analyse numérique et l'optimisation constituent deux aspects importants et souvent complémentaires des mathématiques de l'ingénieur. Une connaissance de notions de base dans ces deux domaines est indispensable pour une ingénierie de bon niveau. Les développements rapides de l'informatique ont fait de ces branches des outils universellement utilisés en technologie et/ou en économie. De nombreux logiciels utilisent divers algorithmes performants d'analyse numérique et d'optimisation, mais avant de pouvoir les utiliser, il faut déjà avoir conscience de leur existence, des problèmes qu'ils peuvent résoudre, avec leurs performances et leurs limitations.

En ignorant la première étape de modélisation, on peut retenir la définition suivante pour l'optimisation:

Définition 1.1

L'optimisation est l'étude de techniques permettant de chercher les minima et/ou les maxima de fonctions.

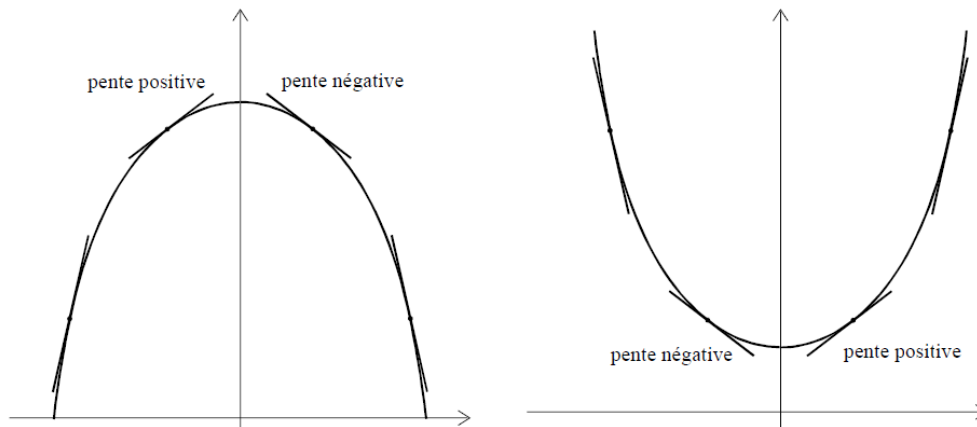
Éléments introductifs à l'optimisation:

En pratique, on rencontre deux types de problèmes : l'optimisation sans contrainte et l'optimisation avec contraintes. Dans les deux cas, le but consiste à trouver les valeurs qui maximisent ou minimisent une fonction objective. Toutefois, dans l'optimisation avec contraintes, les solutions sont soumises à des restrictions (contraintes).

Soit $f(x)$ une fonction d'une variable réelle. Si la fonction $f(x)$ et sa dérivée $f'(x)$ sont continues en un point où la fonction devient décroissante, alors elle possède un maximum. En d'autres termes, la pente de la tangente passe du positif au négatif. Le raisonnement contraire est valable pour un minimum. Dans les deux cas, il s'agit d'un point où la pente de la tangente change de signe. Comme la pente de la tangente est donnée par la première dérivée de la fonction, il faut annuler la première dérivée de f pour obtenir les points candidats (points où la fonction est susceptible de présenter un minimum ou un maximum). Pour chaque point candidat, il faut déterminer s'il s'agit d'un minimum ou d'un maximum.

Critère 1:

Si le signe de la dérivée est positif puis devient négatif quand x croît, alors le point candidat est un maximum de la fonction. Si le signe de la dérivée est négatif puis devient positif quand x croît, alors le point candidat est un minimum de la fonction (voir figure ci-dessous).



Maximum et minimum d'une fonction à une variable

Critère 2:

Le second critère fait appel à la dérivée seconde de la fonction. La dérivée seconde $f''(x)$ d'une fonction est la dérivée de la dérivée première $f'(x)$. Elle mesure le taux de croissance ou de décroissance de $f'(x)$, autrement dit le taux de croissance ou de décroissance de la pente de la tangente à la courbe. Le signe de $f''(x)$ au point candidat $x = x_0$ fournit les renseignements nécessaires. Si $f''(x_0)$ est positive, la pente de la tangente croît quand x croît en passant par x_0 . Inversement, si $f''(x_0)$ est négative, la pente de la tangente décroît quand x croît en passant par x_0 . D'où le résultat suivant basé sur la dérivée seconde.

Résultat :

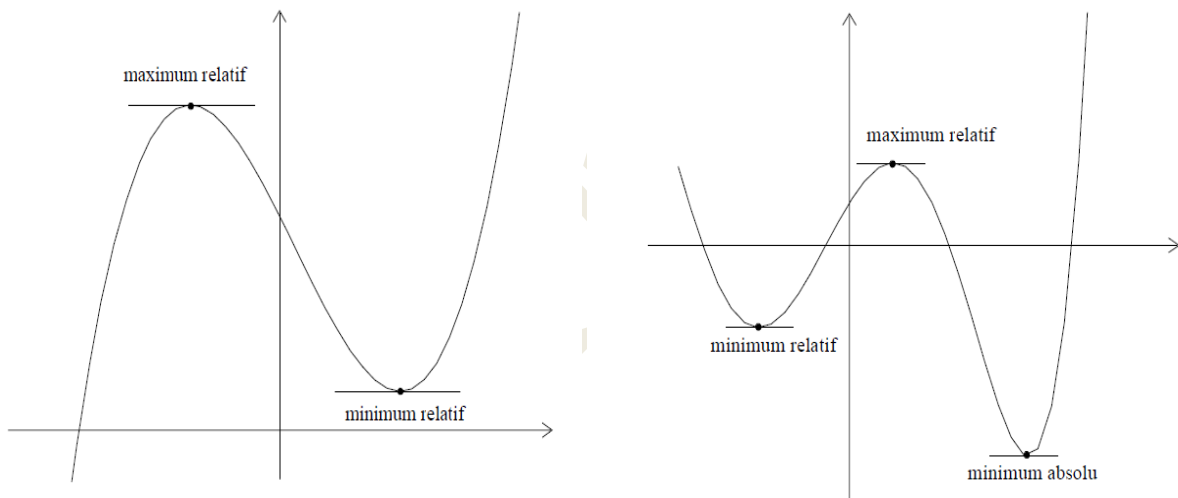
Soit $P(x_0; f(x_0))$ le point en lequel : $f'(x_0) = 0$.

Alors si en ce point :

- ✓ $f'(x_0) < 0$, il s'agit d'un maximum.
- ✓ $f'(x_0) > 0$, il s'agit d'un minimum.

Si pour une fonction $f(x)$, on a un maximum en x_1 et un minimum en x_2 . Certainement au voisinage du point x_1 , la valeur de la fonction est plus petite que $f(x_1)$. On dit que la fonction possède un maximum local ou relatif au point x_1 . De même au point x_2 , on dit que la fonction possède un minimum local ou relatif. Ces deux valeurs, maximum et minimum, sont appelées des extrema locaux ou relatifs, parce qu'il existe des valeurs de x pour lesquelles la fonction prend des valeurs plus grandes que $f(x_1)$ ou plus petites que $f(x_2)$, comme l'illustre la figure en (a) ci-dessous.

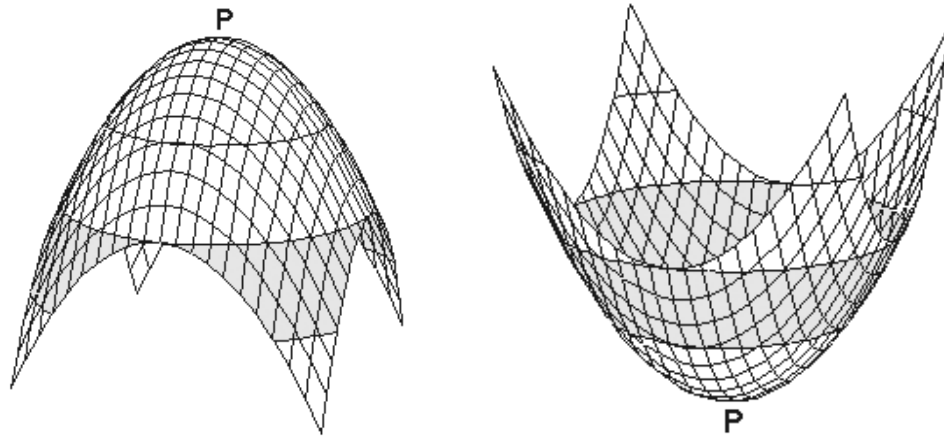
Cependant, on peut définir ce que l'on appelle un maximum ou un minimum global ou absolu. En ce point, la fonction prend la valeur la plus grande (ou la plus petite) sur un intervalle donné ; la figure en (b) ci-dessous montre la différence entre extremums relatifs et extrémum global ou absolu.

**Extrema locaux et absolus**

Considérons à présent le cas des fonctions à plusieurs variables indépendantes.

Des exemples simples en sont fournis par des formules de mathématiques élémentaires. Ainsi, dans la formule de calcul du volume V d'un cylindre droit à base circulaire, $V = \pi r^2 h$, V est une fonction à deux variables indépendantes : r (rayon du cercle de base) et h (hauteur). De la même manière, dans la formule qui donne l'aire A d'un triangle quelconque, $A = \frac{1}{2} xy \sin \alpha$ si, A est une fonction à trois variables indépendantes, x , y et α , qui traduisent respectivement la longueur de deux côtés du triangle et l'angle formé par ces deux côtés.

Pour les fonctions à deux variables, $z = f(x, y)$, le graphe est une surface dans l'espace à trois dimensions. Une telle fonction présente un maximum au point $P(x_0; y_0; f(x_0, y_0))$, si $f(x_0, y_0)$ atteint une valeur supérieure à toutes celles que prend $f(x, y)$ au voisinage de $x = x_0$ et $y = y_0$, comme indiqué sur la figure (a) ci-dessous. De même, $f(x, y)$ possède un minimum au point $P(x_0, y_0; f(x_0, y_0))$, si $f(x_0, y_0)$ atteint une valeur inférieure à toutes celles que prend $f(x, y)$ au voisinage de $x = x_0$ et $y = y_0$; ce cas est illustré par la figure (b) ci-dessous.



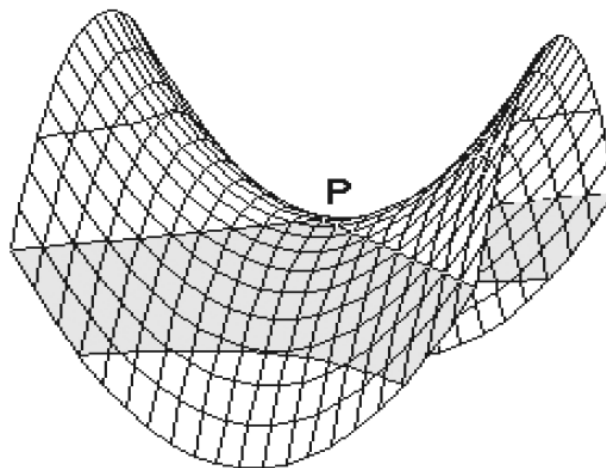
Il en résulte qu'au point $P(x_0; y_0; f(x_0, y_0))$, il existe un plan tangent horizontal. Ce plan tangent est engendré par deux tangentes, elles-mêmes déterminées par :

$$\frac{\partial f}{\partial x} \text{ et } \frac{\partial f}{\partial y}$$

Ainsi, la condition nécessaire à l'existence d'un extremum est la suivante

$$\frac{\partial f}{\partial x} = 0 \text{ et } \frac{\partial f}{\partial y} = 0$$

Cette condition est nécessaire mais pas suffisante. En effet, il existe des fonctions pour lesquelles $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$ sans qu'il existe un extremum en ce point. Dans ce cas, on parle de point-selle. Bien que les deux tangentes soient horizontales, il est toujours possible de trouver un point situé au-dessus du point-selle et un autre au-dessous, ceci quelque soit le voisinage du point selle considéré. Notons encore, qu'en un point-selle la fonction présente un minimum pour l'une des variables et un maximum pour l'autre variable. La figure suivante illustre cette situation.



On aura donc une condition suffisante pour les fonctions à deux variables

En notant $f_{xx} = \frac{\partial^2 f}{\partial x^2}$, $f_{yy} = \frac{\partial^2 f}{\partial y^2}$ et $f_{xy} = \frac{\partial^2 f}{\partial xy}$.

$$\alpha = f_{xx} \cdot f_{yy} - (f_{xy})^2 > 0$$

Résultat:

Soit $P(x_0; y_0; f(x_0, y_0))$ le point en lequel :

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$$

Alors si en ce point :

1. $f_{xx} > 0$ et $\alpha > 0$, f possède un minimum au point P .
2. $f_{xx} < 0$ et $\alpha > 0$, f possède un maximum au point P .
3. $\alpha < 0$, f ne possède ni minimum ni maximum au point P , mais un point-selle.
4. $\alpha = 0$, on ne peut pas conclure.

Notons que cette condition suffisante provient d'un résultat plus général concernant les fonctions à plusieurs variables $f(x_1, x_2, \dots, x_n)$ qu'on va les étudier après.

Formulation mathématique d'un problème d'optimisation:

La forme standard d'un programme mathématique s'exprime par le problème d'optimisation sous contraintes suivant:

$$\text{Minimiser } f(x) \text{ tel que } \begin{cases} g(x) \leq 0, & i = 1, 2, \dots, p \\ h(x) = 0, & i = 1, 2, \dots, q \\ x \in \mathbb{R}^n \end{cases} \equiv \text{Minimiser } f(x) \text{ tel que } x \in D$$

Où $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ et le domaine D est défini par:

$$D = \{x \in \mathbb{R}^n \mid g(x) \leq 0, \quad i = 1, 2, \dots, p \text{ et } h(x) = 0, \quad i = 1, 2, \dots, q\}$$

- ❖ $f: \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction de plusieurs variables ($x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$) à valeurs réelles. Cette fonction qu'on doit minimiser est appelée indifféremment fonction coût, objectif ou critère.
- ❖ $g: \mathbb{R}^n \rightarrow \mathbb{R}^p$ est une fonction de plusieurs variables ($x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$) à valeurs dans \mathbb{R}^p , elle a donc p composantes et on peut l'écrire comme:

$$g(x) = \begin{pmatrix} g_1(x) \\ \vdots \\ g_p(x) \end{pmatrix}$$

Chaque fonction g_i étant définie sur \mathbb{R}^n et à valeurs dans \mathbb{R}^p , la fonction $g(x)$ représente les contraintes en inégalité. La notation $g(x) \leq 0$ signifie qu'on considère les inégalités composante par composante.

- ❖ $h: \mathbb{R}^n \rightarrow \mathbb{R}^q$ est une fonction de plusieurs variables ($x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$) à valeurs dans \mathbb{R}^q , elle a donc q composantes et on peut l'écrire comme:

$$h(x) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_q(x) \end{pmatrix}$$

Chaque fonction h_i étant définie sur \mathbb{R}^n et à valeurs dans \mathbb{R}^q , la fonction $h(x)$ représente les contraintes d'égalité. La notation $h(x) = 0$ signifie qu'on considère les égalités composantes par composante.

\mathbb{R}^n définit le domaine de recherche des variables, cet ensemble définit l'espace d'état, tant dis que l'ensemble de points de l'espace des états possibles qui satisfait au mieux les contraintes est donné par:

$$D = \{x \in \mathbb{R}^n \mid g(x) \leq 0, \quad i = 1, 2, \dots, p \text{ et } h(x) = 0, \quad i = 1, 2, \dots, q\}$$

Il est possible de passer d'un problème de minimisation à un problème de maximisation grâce à la propriété suivante:

$$\max_{x \in D} f(x) = - \min_{x \in D} (-f(x))$$

En faite, soit x^* tel que $f(x^*) = \min\{f(x), x \in D\}$, On a donc:

$$\begin{aligned} \forall x \in D, \quad f(x) &\geq f(x^*) \\ \Leftrightarrow \forall x \in D, -f(x) &\leq -f(x^*) \\ \Leftrightarrow -f(x^*) &= \max\{-f(x) \mid \forall x \in D\} \\ \Leftrightarrow f(x^*) &= -\max\{-f(x) \mid \forall x \in D\} \end{aligned}$$

Classification des problèmes d'optimisation

Suivant la nature de la fonction objectif $f(x)$, des contraintes $g(x)$ et des variables de décision x , le problème d'optimisation porte des noms divers. On distingue plus particulièrement les cas suivants:

- ❖ La programmation linéaire:

Comme son nom l'indique, la fonction objectif et les contraintes sont toutes linéaire et $D = \mathbb{R}^n$.

- ❖ La programmation quadratique

Lorsque la fonction objectif est quadratique et les contraintes son toutes linéaires et $D = \mathbb{R}^n$.

- ❖ La programmation convexe

C'est le cas d'optimisation où la fonction objectif et les contraintes d'inégalités sont convexes avec des contraintes d'égalités linéaires et un domaine D convexe.

- ❖ De même selon la nature (discrète ou continue) des variables de décision x , on parle de programmation mathématique (optimisation) discrète ou continue.
- ❖ On peut aussi classier la nature d'optimisation selon que la fonction critère est mono-objective et on aura un problème d'optimisation mono-objectif ou qu'elle est multi-critère (un compromis entre plusieurs objectifs contradictoires est recherché), et on parlera d'optimisation multi-objectif.

Remarque:

Rappelons-nous que toutes les méthodes d'optimisation recherchent un point ou un ensemble de points dans l'espace de recherche (domaine D) qui minimisent ou maximisent la fonction objective. Dans le cas de problème d'optimisation général (fonction objective et contraintes quelconques), le problème est très difficile à résoudre (voir impossible) ou prend un très long temps de calcul sans certitude de trouver une solution optimale à moins qu'on est dans des situations de structures spéciales. Heureusement, beaucoup de problèmes rencontrés peuvent être exprimés sous la forme de problème d'optimisation convexe plus attaquable (solvable).

Un vecteur \bar{x} vérifiant les contraintes d'un PM est dit solution ou solution réalisable d'un PM. L'ensemble de solutions réalisables d'un PM forme son domaine de définition. Celui-ci peut être vide (dans ce cas le problème n'admet pas de solution), dans le cas contraire, on dit le PM admet des solutions. Sous certaines conditions, il peut exister des solutions x^* dites optimales, c'est-à-dire qui maximisent ou minimisent la fonction objectif sur toutes les solutions \bar{x} .

Définition 1.2 (Minimum local):

Soit D un domaine non vide de \mathbb{R}^n et f une fonction de D dans \mathbb{R} , on dit que $x^* \in D$ réalise un minimum local de f sur D , si on peut trouver une boule $B(x^*)$ centrée en x^* telle que:

$$\forall x \in D \cap B(x^*) \Rightarrow f(x^*) \leq f(x)$$

Sachant qu'une boule centrée en x^* et de rayon ε est donnée par l'ensemble des points vérifiant:

$$B(x^*, \varepsilon) = \{x \in D / \|x - x^*\| \leq \varepsilon\}$$

Définition 1.3 (Minimum global):

Soit D un domaine non vide de \mathbb{R}^n et f une fonction de D dans \mathbb{R} , on dit que $x^* \in D$ réalise un minimum global de f sur D , Si : $\forall x \in D \Rightarrow f(x^*) \leq f(x)$.

Et on écrit:

$$f(x^*) = \min_{x \in D} f(x) \Leftrightarrow x^* = \operatorname{argmin}_{x \in D} f(x)$$

Les conditions d'existence et d'unicité de solution optimale se fait à travers une étude des fonctions objectives ainsi que les domaines admissibles des variables de décision. Par exemple une fonction multimodale présente plusieurs minima tant dis que une fonction unimodale n'a qu'un minimum, le minimum global. Donc la différentiabilité et la convexité sont des propriétés importantes à investiguer.

Avant d'aborder ce volet mathématique, nous allons voir comment peut-on rencontrer ce genre de problème en pratique à commencer par des exemples simples avec ou sans contraintes pouvant faire l'objet d'optimisation au sens des moindres carrés ou de programmation linéaire toutes les deux représentent des cas particuliers de l'optimisation convexe.

A. Méthode des moindres carrés:

La méthode des moindres carrés est très utilisée dans les sciences expérimentales et plus particulièrement dans les problèmes d'estimation et d'identification. Comme son nom l'indique, cette méthode consiste à minimiser la norme quadratique (norme euclidienne) d'une fonction appelée fonction d'erreur. Le problème à résoudre peut s'énoncer de différentes manières mais optent toujours pour la résolution d'un système d'équations de la forme:

$$Ax = b.$$

Où A est une matrice quelconque à m lignes et n colonnes donc $A \in \mathbb{R}^{m \times n}$ et les deux vecteurs $x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$.

Plusieurs situations peuvent être rencontrées selon la structure de la matrice A et les valeurs de m et n .

1) Cas 1 : $m = n$ (A une matrice carrée donc inversible):

Si on note $M_n(\mathbb{R})$ l'ensemble des matrices carrées d'ordre n , soit $A \in M_n(\mathbb{R})$ une matrice inversible, et soit $b \in \mathbb{R}^n$. On considère le système d'équations linéaires de Cramer :

$$\begin{cases} Ax = b \\ x \in \mathbb{R}^n \end{cases}$$

La résolution d'un tel système intervient très fréquemment dans tous les domaines d'applications mathématiques. Lorsque n est grand la résolution directe de ce système (par la méthode du pivot de Gauss, ou par les formules de Cramer, par exemple) est fastidieuse et prend un temps de calcul pouvant être pénalisant. Ainsi, il est très utile dans la pratique de disposer d'algorithmes de résolution approchée de système d'équations linéaires, plus rapides et/ou moins gourmands en ressources.

Pour ce faire, il faut d'abord se ramener au cas d'une matrice symétrique définie positive. Il suffit de multiplier à gauche par la matrice transposée A^t , c'est une opération qui ne demande pas beaucoup de calcul.

Posons $M = A^t A$ et $N = A^t b$ et on aura M une matrice symétrique définie positive, on obtient le système linéaire suivant:

$$Mx = N$$

Comme M est inversible, donc inverser la matrice M nous permettra de résoudre directement le système ($x = M^{-1}N$). Sachant que l'inversion d'une matrice symétrique définie positive est une opération coûteuse numériquement. Vu que cet opération d'inversion de matrice intervient fréquemment en mathématiques appliquées, plusieurs méthodes ont été développées pour l'effectuer au mieux, citons par exemple la méthode du pivot de Gauss, ou la factorisation LU, qui toutes deux se ramènent à l'inversion de matrices triangulaires (opération simple). Mais si la dimension du système précédent est grande, on préfère les méthodes de résolution approchées même s'il y a une possibilité de résolution exacte.

NB

On verra plus loin dans le cours que résoudre le système linéaire $Ax = b$ est équivalent au problème d'optimisation quadratique ci-dessous:

$$\text{Minimiser } f(x) \text{ tel que } \begin{cases} f(x) = \frac{1}{2} x^t A x - b^t x \\ x \in R^n \end{cases}$$

Ainsi une méthode itérative de recherche de minimum de ce problème d'optimisation quadratique fournit une méthode de résolution approchée du système linéaire $Ax = b$.

2) Cas 2 : $m > n$ (A une matrice rectangulaire non inversible):

Dans ce cas, il ya plus d'équations que d'inconnus, le problème est surdéterminé (en général n'admet pas de solution) et on doit chercher une solution approchée qui vérifie la consistance du système d'équations selon un certain critère d'erreurs.

Une des vieilles méthodes pour l'approximation de solution de tel système est de définir une erreur résiduelle $r = Ax - b$ et chercher une solution x_{ls} qui minimise la norme $\|r\|$. Cette solution x_{ls} est appelée solution (approchée) au sens des moindres carrés du système $Ax = b$.

Pour cela, nous supposons que A est de rang plein (donc égal à n), puis nous minimisons le carré de la norme de l'erreur résiduelle:

$$\|r\|^2 = x^t A^t A x - 2b^t A x + b^t b$$

$$\text{Or } \nabla_x \|r\|^2 = 2A^t A x - 2A^t b$$

$$\text{Et } \nabla_x \|r\|^2 = 0 \Rightarrow A^t A x = A^t b$$

C'est l'équation normale qui utilise le fait que $A^t A$ est inversible, on aura la solution approchée:

$$x = (A^t A)^{-1} A^t b$$

Avec $(A^t A)^{-1} A^t$ pseudo-inverse de la matrice A .

Comme c'est déjà signalé, calculer l'inverse d'une matrice demande plus d'effort de calcul et plus de temps d'autant plus pour les pseudo-inverses. C'est pourquoi on fera généralement recours aux techniques de factorisation de matrices.

Un exemple de factorisation QR: consiste à chercher deux matrices Q et R telle que: $A=QR$, $A \in \mathbb{R}^{m \times n}$ est de rang plein, $Q^t Q = I_n$ et R une matrice triangulaire supérieur inversible.

Et on aura:

$$(A^t A)^{-1} A^t = (R^t Q^t Q R)^{-1} R^t Q^t = R^{-1} Q^t \Rightarrow R x = Q^t b$$

Ainsi, la résolution de $Ax = b$ peut être faite en résolvant $Rx = Q^t b$ qu'on peut facilement effectuer par substitution puisque R est triangulaire.

3) Cas 3 : $m < n$ (A une matrice rectangulaire non inversible):

Si $m < n$, alors le système $Ax = b$ est sous déterminé donc possédant une infinité de solution lorsque le rang de la matrice A est plein (donc égal à m). L'ensemble de solution est un espace affine de dimension $(n-m)$. En effet, on peut décomposer la matrice A (par permutation de colonnes) sous la forme $A = [A_1 \ A_2]$ telque A_1 est une matrice carré et de même pour $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, alors notre système s'écrit:

$$A_1 x_1 + A_2 x_2 = b, \quad \text{et on aura:}$$

$x_1 = A_1^{-1} b - A_1^{-1} A_2 x_2$, et l'ensemble des solutions est l'espace affine de dimension $n-m$ défini par:

$$S = \left\{ u \in \mathbb{R}^n \mid u = \begin{pmatrix} A_1^{-1} b \\ 0 \end{pmatrix} + \begin{pmatrix} -A_1^{-1} A_2 \\ I \end{pmatrix} y, y \in \mathbb{R}^{n-m} \right\}$$

Parmi toutes ces solutions, on est souvent amené à choisir une solution particulière. On choisit alors souvent de considérer la solution de norme quadratique minimale comme précédemment et on aura aussi une solution de la forme:

$$x = A^t b (A^t A)^{-1}$$

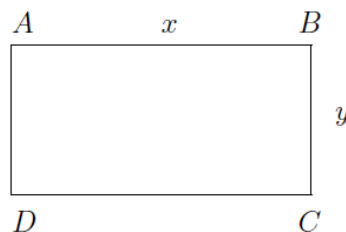
4) Cas 4 : Cas général: ($m \neq n$ et rang de la matrice A est \neq de m et de n):

Dans le cas général la matrice A du système $Ax = b$ n'est pas forcément de rang plein. On peut se ramener à un système de rang plein de diverses façons. Par exemple, en faisant une décomposition en valeurs singulières de A.

Exemples d'optimisation en pratique:

Exemple 1:

Considérons un rectangle ABCD. On appelle x la longueur AB et y, la longueur BC. On a évidemment : $x > 0$ et $y > 0$.



On appelle $P(x, y)$, le périmètre de ABCD, $S(x, y)$, l'aire de ce rectangle.

On a alors :

$$P(x, y) = 2(x + y)$$

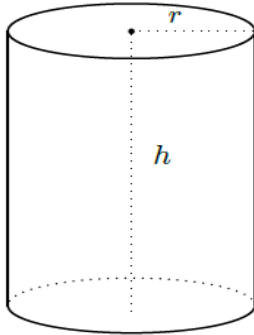
$$S(x, y) = xy.$$

Problème courant en Optimisation : on peut être amené à chercher x pour que le périmètre de ABCD soit minimal sachant que son aire vaut 1.

Formulation mathématique: Minimiser $P(x, y)$ tel que $\begin{cases} S(x, y) = 1 \\ x > 0 \\ y > 0 \end{cases}$

Exemple 2: Minimisation des coûts dans la fabrication de boîtes cylindriques

Dans la fabrication de boîtes de conserve cylindriques on minimise les coûts de matière première en cherchant le cylindre de surface minimale à volume constant. Considérons un cylindre, donné par sa hauteur h et le rayon r de sa base.



Le volume est : $V = \pi r^2 h = K = \text{constante}$.
 L'aire est : $S = 2\pi r^2 + 2\pi r h$.
 Le problème d'optimisation s'écrit :

Minimiser $S(r, h)$ tel que $\begin{cases} V(r, h) = K \\ r > 0 \\ h > 0 \end{cases}$

On utilise la contrainte égalitaire pour se ramener à un problème à une variable :

$$h = \frac{K}{\pi r^2} \Rightarrow S(r) = 2\pi r^2 + 2 \frac{K}{r}$$

et notre problème d'optimisation devient:

$$\begin{cases} \text{Min}(S(r)) = \text{Min}\left(\pi r^2 + \frac{K}{r}\right) \\ r > 0 \end{cases}$$

On étudie les variations de $S(r)$:

$$\frac{dS(r)}{dr} = \frac{2\pi r^3 - K}{r^2} \geq 0 \Rightarrow r \geq \sqrt[3]{\frac{K}{2\pi}}$$

r	0	$\sqrt[3]{\frac{K}{2\pi}}$	$+\infty$
S'	-	0	+
S			

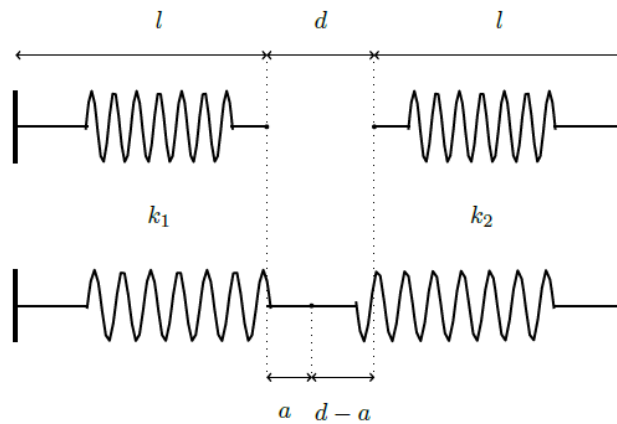
Le minimum est obtenu pour: $r_{\min} = \sqrt[3]{\frac{K}{2\pi}}$

Alors $h_{\min} = \frac{K r_{\min}}{\pi (r_{\min})^3} = \sqrt[3]{\frac{4K}{\pi}} = 2r_{\min}$

Et on obtiendra $S_{\min}(r_{\min}, h_{\min}) = 6 \sqrt[3]{\frac{\pi K^2}{4}}$

Exemple 3: Position d'équilibre d'un système de deux ressorts.

Deux ressorts de coefficients de tension k_1 , k_2 et ayant même longueur à vide, ont chacun une extrémité fixe, et l'autre à distance mutuelle d . Lorsqu'on les attache par leur extrémité libre, comment s'exprime leur position d'équilibre (figure ci-dessous) ?



L'énergie potentielle à l'équilibre du premier ressort est:

$$E_1 = \frac{1}{2} k_1 a^2$$

L'énergie potentielle à l'équilibre du deuxième ressort est:

$$E_2 = \frac{1}{2} k_2 (a - d)^2$$

L'énergie totale du système est:

$$E_1 + E_2 = \frac{1}{2} (k_1 a^2 + k_2 (a - d)^2)$$

La position d'équilibre est celle pour laquelle l'énergie potentielle du système est minimale.

Il s'agit donc d'un problème d'optimisation qui s'exprime :

$$\begin{cases} \text{Min}(E(a)) = k_1 a^2 + k_2 (d - a)^2 \\ 0 \leq a \leq d \end{cases}$$

De même que précédemment, en étudiant les variations de $E(a)$, on trouvera que la position d'équilibre est atteinte pour:

$$d - a = \frac{k_1}{k_1 + k_2} d$$

Exemple 4: Problème de transport

On considère un problème de ravitaillement très utile en pratique; il fait partie d'une large classe de problème (largement étudiée que ce soit en optimisation continue ou en optimisation combinatoire, et que l'on sait résoudre efficacement).

On souhaite approvisionner en carburant 3 sites à partir de 2 dépôts de capacité limitée.

L'acheminement en carburant d'un dépôt à un site a un coût unitaire. Le tableau suivant résume chacun de ces coûts ainsi que la demande de chaque site, et le stock disponible dans chaque dépôt.

	site 1	site 2	site 3	diponibilité
dépôt 1	10	12	9	300
dépôt 2	11	11	10	450
demande	200	250	250	

Notons $x_{i,j}$ $i=1, 2$ et $j=1, 2, 3$ la quantité de carburant (en unité de volume) acheminé du dépôt i au site j . Le coût d'acheminement est donné par la fonction de coût suivante :

$$f(x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}) = 10x_{11} + 12x_{12} + 9x_{13} + 11x_{21} + 11x_{22} + 10x_{23}$$

qu'il s'agit de minimiser la fonction de coût, sous les contraintes :

❖ Contraintes égalitaires provenant de la demande :

$$\begin{cases} x_{11} + x_{21} = 200 \\ x_{12} + x_{22} = 250 \\ x_{13} + x_{23} = 250 \end{cases}$$

❖ Contraintes inégalitaires provenant du stock disponible :

$$\begin{cases} x_{11} + x_{12} + x_{13} \leq 300 \\ x_{21} + x_{22} + x_{23} \leq 450 \end{cases}$$

❖ Contraintes de signe :

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23} > 0.$$

Il s'agit là d'un problème d'optimisation encore appelé problème de programmation linéaire.

Exemple 5: Modélisation de données expérimentales (Identification)

Plus généralement supposons que l'on ait effectué une série de p mesures dépendant d'un paramètre (évolution d'une grandeur physique, en fonction du temps, etc...).

Instants de mesure	t_1	t_2	t_p
Valeur mesurée	y_1	y_2	y_p

On souhaite modéliser ces données expérimentales par une certaine fonction mathématique f : de \mathbb{R} dans \mathbb{R} dépendant de paramètres réels $(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n)$.

On cherche à déterminer les paramètres pour lesquels les valeurs prises par la fonction aux instants $t_1, t_2, t_3, \dots, t_p$ collent au mieux aux valeurs mesurées, dans un certain sens, disons par exemple au sens des moindres carrés.

Il s'agit alors bien d'un problème d'optimisation :

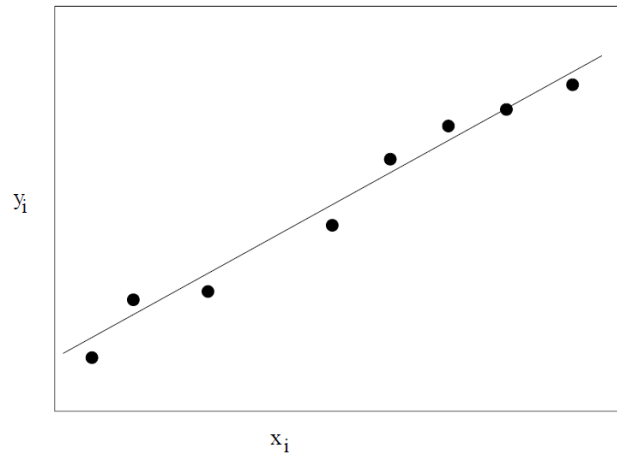
$$\min_{(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n)} \sum_1^p (y_i - f_{(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n)}(t_i))^2$$

1^{er} cas de figure: Régression linéaire

Supposons donnés N points d'un relevé expérimental $(x_i; y_i)$; $(i = 1; \dots; N)$: x_i est par exemple l'instant de prélèvement d'une substance dans une réaction chimique et y_i est la concentration de cette substance.

Supposons maintenant que l'on recherche une relation linéaire entre x et y de la forme:

$$y = ax + b \text{ (autrement dit on cherche à approximer ce nuages de points par une droite)}$$



Il est clair que pour $N = 2$ les inconnues a et b sont déterminés de manière unique. Il est clair aussi que la droite ne peut passer par l'ensemble des points (voir figureci-dessus). En fait ce qu'on recherche ici est la droite la plus probable, donc minimisant une certaine erreur.

Au point $(x_i; y_i)$; la distance entre la droite et le point est égale à:

$$e_i = |(ax_i + b) - y_i|$$

On peut alors envisager de déterminer a et b en résolvant différents problèmes d'optimisation:

- ❖ minimiser la plus grande valeur de e_i (problème de min-max)

$$\min_{a,b} \max_i e_i$$

- ❖ minimiser la somme des erreurs (programmation linéaire)

$$\min_{a,b} \sum_0^{n-1} e_i$$

- ❖ minimiser la somme des carrés des erreurs (droite des moindres carrés)

$$\min_{a,b} \sum_0^{n-1} e_i^2$$

Ceci est équivalent à:

$$\min_{a,b} e^t e = \min_{a,b} \|e\|_2^2 \quad \text{Avec} \quad e = \begin{pmatrix} e_0 \\ \vdots \\ e_{n-1} \end{pmatrix}$$

Vue l'intérêt et la facilité de manipulation de la norme 2, on préfère toujours la dernière forme d'optimisation (voir développement ci-après).

Il est possible d'effectuer le calcul précédent sous forme vectoriel:

On pose $f(a,b) = e^t e = \sum_0^{n-1} e_i^2$, $e_i = (ax_i + b) - y_i$

On pose $e = Cz - d$

Avec

$$C = \begin{pmatrix} x_0 & 1 \\ x_1 & 1 \\ \vdots & \vdots \\ \vdots & \vdots \\ x_{n-1} & 1 \end{pmatrix}, \quad z = \begin{pmatrix} a \\ b \end{pmatrix}, \quad d = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

On aura:

$$f(a, b) = e^t e = z^t C^t C z - 2(C^t d)^t z + d^t d$$

Or

$$\frac{\partial f}{\partial z} = 2(C^t C)z - 2C^t d$$

Les paramètres a et b de la régression linéaire sont donnés par:

$$\frac{\partial f}{\partial z} = 0$$

C.à.d.

$$\begin{pmatrix} a \\ b \end{pmatrix} = (C^t C)^{-1} C^t d$$

NB

Cet équation est très générale, elle reste valable dans le cas de n paramètres $z = \begin{pmatrix} z_1 \\ \vdots \\ z_{n-1} \end{pmatrix}$, $n \leq N$ avec $C : (N \times n)$ et $d : (N \times 1)$.

2^{ème} cas de figure: Régression parabolique

On se donne un ensemble de m points $(x_i; y_i)$. On cherche les coefficients p, q, r de la parabole:

$$y = px^2 + qx + r$$

qui passe "au plus près" des points. On suppose qu'on dispose d'un nombre suffisant de points $m \geq 3$. Que veut-on dire exactement par "au plus près"? Pour chaque triplet de valeurs p, q, r, on considère les m écarts verticaux entre la courbe et les points $(x_i; y_i)$

$$\text{écart}_i = px_i^2 + qx_i + r - y_i, \quad 1 \leq i \leq m$$

Et on définit l'erreur comme la somme des carrés des écarts :

$$\text{erreur} = \text{écart}_1^2 + \text{écart}_2^2 + \dots + \text{écart}_m^2$$

On cherche les coefficients p, q, r pour lesquels l'erreur (la somme des carrés des écarts verticaux) est minimale. Pour ce faire, on pose que la parabole passe par les m points :

$$px_1^2 + qx_1 + r = y_1$$

$$px_2^2 + qx_2 + r = y_2$$

⋮

$$px_m^2 + qx_m + r = y_m$$

Les équations ci-dessus forment un système d'équations linéaires qui n'a pas de solution en général :

$$\underbrace{\begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_m^2 & x_m & 1 \end{pmatrix}}_C \underbrace{\begin{pmatrix} p \\ q \\ r \end{pmatrix}}_z = \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}}_d$$

Et on trouvera notre fameux système linéaire qui peut se mettre sous $e = Cz - d$, qu'on sait résoudre efficacement par la méthode des moindres carrés.

Extension de la méthode:

En réalité la méthode des moindres carrés précédemment développée reste applicable du moment que la fonction d'approximation recherchée dépend de manière linéaire des inconnues. C'est le cas de toute combinaison linéaire de fonctions non-linéaires préalablement choisis.

Exp: Pour la fonction:

$$y = a \ln x + b \cos x + ce^x$$

Les matrices du calcul vectoriel prennent la forme suivante:

$$C = \begin{pmatrix} \ln x_0 & \cos x_0 & e^{x_0} \\ \ln x_1 & \cos x_1 & e^{x_1} \\ \vdots & \vdots & \vdots \\ \ln x_{n-1} & \cos x_{n-1} & e^{x_{n-1}} \end{pmatrix}, z = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, d = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

3^{ème} cas de figure: Convolution

On considère une équation de convolution de la forme $y = h * x$, où h représente la réponse impulsionnelle d'un filtre, x son entrée et y sa sortie. L'échantillonnage de l'équation intégrale de convolution $y = h * x$ donnée par $y = \int_0^{T_h} h(u)x(t-u)du$ (T_h est la durée de la réponse impulsionnelle du filtre causal h) conduit à l'écriture $y_n = \sum_0^L h_k x_{n-k}$ où $x_n = x(nT)$ et $h_n = T^{-1}h(nT)$ sont obtenues par échantillonnage de l'équation de convolution avec un pas égal à T .

❖ **1^{ère} situation:**

On suppose dans un premier temps que le filtre h est connu mais que le signal x est inconnu. On cherche à retrouver les valeurs de x_n, \dots, x_{n+N} à partir de l'observation de y sur le même intervalle de temps, c'est à dire l'observation de y_n, \dots, y_{n+N} .

Les relations de convolution s'écrivent

$$y_{n+k} = \sum_{j=0}^L h_k x_{n+k-j}, \quad k = 0, \dots, N$$

- En considérant l'étalement temporel du filtre h et $N > L$, on aura à faire donc à un vecteur

d'entrée $x = \begin{pmatrix} x_{n-L} \\ x_{n-L+1} \\ \vdots \\ x_n \\ \vdots \\ x_{n+N} \end{pmatrix}$ ce qui augmente le nombre d'inconnus. Ainsi, le système est sous-

déterminé car il fait intervenir plus d'inconnues que d'équations. Si on opte pour les moindres carrés pour la résolution du système alors on se retrouve dans le cas 3.

Mise sous forme matricielle: $y=Hx$

$$\begin{pmatrix} y_n \\ y_{n+1} \\ \vdots \\ y_{n+N} \end{pmatrix} = \begin{pmatrix} h_L & h_{L-1} & \dots & h_0 & 0 & \dots & 0 \\ 0 & h_L & h_{L-1} & \dots & 0 & \dots & 0 \\ \vdots & 0 & & & & & \\ 0 & \dots & 0 & h_L & h_{L-1} & \dots & h_0 \end{pmatrix} \begin{pmatrix} x_{n-L} \\ x_{n-L+1} \\ \vdots \\ x_n \\ \vdots \\ x_{n+N} \end{pmatrix}$$

La décomposition de H et x conduit à

$$y = H_1x_1 + H_2x_2, \quad \text{avec } H_1 \text{ inversible}$$

Avec $x_1 = \begin{pmatrix} x_n \\ \vdots \\ x_{n+N} \end{pmatrix}$, $x_2 = \begin{pmatrix} x_{n-L} \\ \vdots \\ x_{n-1} \end{pmatrix}$ on aura $H_1 = \begin{pmatrix} h_0 & 0 & \dots & \dots & \dots & \dots & 0 \\ h_1 & h_0 & 0 & \dots & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_L & h_{L-1} & \dots & h_0 & 0 & \dots & 0 \\ 0 & h_L & h_{L-1} & \dots & h_0 & 0 & \dots \\ \vdots & 0 & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & h_L & h_{L-1} & \dots & h_0 \end{pmatrix}$ et

$$H_2 = \begin{pmatrix} h_L & h_{L-1} & \dots & h_1 \\ 0 & h_L & \dots & h_2 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & h_L \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

Notre solution sera donnée par:

$$x = x_1 = H_1^{-1}y - H_1^{-1}H_2x_2$$

Une solution bien paramétrée par le vecteur x_2

- En l'absence de l'étalement temporel du filtre h , on a $x_2 = 0$ et on obtient un système facilement solvable $y = H_1x_1 \rightarrow x = x_1 = H_1^{-1}y$ pour (puisque H_1 triangulaire)

❖ 2^{ème} situation:

On peut se mettre dans la situation où le signal d'entrée x est connu tant dis que les coefficients du filtre h sont maintenant inconnus, et l'on observe les sorties y_m en présence d'un bruit d'observation v , dans ce cas on a: $Y=Xh+v$

$$y_m = \sum_{k=0}^L h_k x_{m-k} + v_m, \quad m = n, n + 1, \dots, n + N$$

Qui peut se mettre sous forme matricielle (toujours pour $N>L$) comme suit:

$$\begin{pmatrix} y_n \\ y_{n+1} \\ \vdots \\ y_{n+N} \end{pmatrix} = \begin{pmatrix} x_n & x_{n-1} & \dots & x_{n-L} \\ x_{n+1} & x_n & \dots & x_{n-L+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n+N-L} & x_{n+N-L+1} & \dots & x_{n+N} \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \end{pmatrix} + \begin{pmatrix} v_n \\ v_{n+1} \\ \vdots \\ v_{n+N} \end{pmatrix}$$

Le système comporte alors plus d'équations que d'inconnues, Il est dit surdéterminé. Même lorsque v est nul, il n'est cependant pas évident de savoir a priori pour un système surdéterminé quelles équations peut-on éliminer pour se ramener à un système carré inversible (sous réserve que la matrice intervenant dans la relation initiale soit de rang plein).

A défaut d'une solution exacte, on cherche alors une solution approchée selon par exemple le critère des moindres carrés et on se retrouve dans le cas 2 précédemment étudié.

Définition 1.4 (contrainte active)

Soit une contrainte d'inégalité $g_i(x) \leq 0$ avec $g_i(x)$ une fonction au moins continue de R^n dans R . Si pour un point x^* de R^n on a $g_i(x^*) < 0$, on dira que la contrainte est inactive en x^* . Tandis que si x^* satisfait $g_i(x^*) = 0$, on dira que la contrainte est active ou saturée en x^* et on note $I(x^*)$ l'ensemble des indices j correspondant aux contraintes actives en x^* .

Définition 1.5 (point régulier)

Il existe de nombreux critères de régularité, la plus faible s'énonce: On dit qu'un point $x^* \in \mathbb{R}^n$ est régulier pour les contraintes d'égalités $h(x)$ et d'inégalités $g(x)$ si:

- il est réalisable c.à.d. $h(x^*) = 0$ et $g(x^*) \leq 0$
- les vecteurs $\nabla h_i(x^*), i = 1, \dots, q$. sont indépendants
- on peut trouver un vecteur non nul de \mathbb{R}^n tel que:

$$d^t \nabla h_i(x^*) = 0, i = 1, \dots, q \text{ et } d^t \nabla g_j(x^*) < 0, j \in I(x^*).$$

Les deux dernières conditions représentent la contrainte de qualification du point x^*

Remarque:

Un problème de minimisation n'admet pas nécessairement une solution par exemple: $\min(\frac{1}{x})$ pour $x \in [1, +\infty]$ n'admet pas de solution tandis que $\min(\cos(x))$ pour $x \in [\pi, 4\pi]$ admet deux solutions.

Remarque:

Ce n'est qu'un abus de langage, qu'on dit souvent que x^* est un minimum pour la fonction $f(x)$: il faudrait dire que x^* réalise un minimum de $f(x)$ sur D ou que $f(x^*)$ est le minimum de $f(x)$ sur D .

Évidemment, on ne peut pas résoudre le problème de minimisation en général sans quelques hypothèses permettant d'assurer au moins l'existence de la solution, logiquement ces hypothèses concernent la fonction objectif $f(x)$ et les fonctions de contraintes $g(x)$ et $h(x)$ qui définissent le domaine D . Le cas où les données sont liées par une fonction objective convexe est très important car les problèmes quadratiques (donc convexe) sont les plus rencontrés en engineering et sont à l'origine de la plupart des algorithmes non linéaires. Nous en rappelons quelques définitions et propriétés ci-après.

Nous commençons d'abord par une propriété importante sur l'existence d'extrémums pour des fonctions quelconques (donc pas forcément convexe) définies sur un domaine compact:

Définition 1.6

Si D est un domaine compact (i.e. fermé et borné) de \mathbb{R}^n , et $f: D \rightarrow \mathbb{R}$ est continue, alors f admet un minimum ainsi qu'un maximum global sur D .

Ce résultat n'est utile que pour un problème d'optimisation sous contraintes, car c'est dans ce cas où le domaine est toujours un fermé de \mathbb{R}^n et c'est le seul cas où il peut être fermé, c.à.d. compact.

Définition: (fonction convexe)

On dit qu'une fonction $f: D \subset \mathbb{H} \rightarrow \mathbb{R} \cup \{+\infty\}$ est convexe si D est convexe et si

$$\forall (x, y) \in D \times D, \forall t \in [0, 1] \text{ alors } f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y)$$

Définition: (fonction strictement convexe)

On dit qu'une fonction $f: D \subset \mathbb{H} \rightarrow \mathbb{R} \cup \{+\infty\}$ est strictement convexe si D est convexe et si

$$\forall (x, y) \in D \times D \text{ avec } x \neq y, \forall t \in [0, 1] \text{ alors } f(tx + (1 - t)y) < tf(x) + (1 - t)f(y)$$

Définition 1.7 (domaine d'une fonction convexe)

Soit $f: \mathbb{H} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe, on appelle domaine de f l'ensemble

$$\text{dom } f = \{x \in \mathbb{H} | f(x) < \infty\}$$

cet ensemble est convexe.

Lorsque le domaine de f est non vide, on dit que f est une fonction propre.

Définition: (fonction coercive)

Une application $f: D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ continue est coercive si D est un fermé non borné et si

$$\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$$

Corollaire:

Toute fonction convexe propre sur un espace de dimension finie ($H=\mathbb{R}^n$) est continue sur l'intérieur de son domaine.

Théorème 1.1:

Soit f une fonction continue sur un sous-ensemble D fermé de \mathbb{R}^n . On suppose que

- ou bien D est borné
- ou bien D est non borné mais f est coercive

Alors f possède un minimum sur D .

Théorème 1.2:

Une application coercive admet un minimum global (et aucun maximum global). Si $(-f)$ est coercive alors f admet un maximum global (et aucun minimum global).

Exemple:

Il est évident qu'une fonction polynomiale $f: \mathbb{R} \rightarrow \mathbb{R}$ de degré pair >0 est coercive sur \mathbb{R} si et seulement si le coefficient de son terme de plus haut degré est >0 et aura donc un minimum global. Tandis que si le coefficient de son terme de plus haut degré est <0 , elle aura donc un maximum global. Si le degré de la fonction polynomiale est impair, elle n'a ni maximum ni minimum.

Différentiabilité des fonctions convexes

Définition 1.8

Soit une fonction $f: H \rightarrow \mathbb{R} \cup \{+\infty\}$, on dit que f est Gâteaux-différentiable en $u \in \text{dom } f$ si la dérivée directionnelle

$$f'(u; v) = \lim_{t \rightarrow 0} \frac{f(u + tv) - f(u)}{t}$$

existe dans toute direction v de H et si l'application $v \rightarrow f'(u; v)$ est linéaire continue.

On note alors

$$f'(u; v) = \nabla f(u)^t v$$

Où $\nabla f(u)$ est le gradient de f en u .

Il est clair que si f est différentiable au sens classique en u (Fréchet différentiable), alors f est Gâteaux-différentiable en u et les deux dérivées coïncident; la réciproque n'est pas vraie.

Théorème 1.3:

Soit $f: D \subset H \rightarrow \mathbb{R}$, Gâteaux-différentiable sur D avec D convexe, alors f est convexe si et seulement si

$$\forall (u, v) \in D \times D \quad f(v) \geq f(u) + \nabla f(u)^t (v - u)$$

Théorème 1.4:

Soit $f: D \subset H \rightarrow \mathbb{R}$, Gâteaux-différentiable sur D avec D convexe, alors f est convexe si et seulement si ∇f est un opérateur monotone c.à.d

$$\forall (u, v) \in D \times D \quad \nabla f(u) - \nabla f(v)^t(u - v) \geq 0$$

Évidemment, si ∇f est un opérateur strictement monotone alors f est strictement convexe.

Remarque:

On définit de manière analogue la (Gâteaux) dérivée seconde de f en u , comme étant la dérivée de la fonction vectorielle $u \rightarrow \nabla f(u)$. On la note $D^2f(u)$ et on l'appelle aussi Hessien par abus de langage. Ce Hessien est identifiable à une matrice carrée $n \times n$ lorsque $H = \mathbb{R}^n$.

Dans ce qui suit, nous allons étudier le problème de minimisation en dégagant des conditions d'existence et d'unicité de solutions en terme de conditions nécessaires et/ou suffisantes d'optimalité puis formuler explicitement les algorithmes de calcul de ces solutions.

Chapitre 2

Optimisation Sans Contraintes

Minimisation sans contraintes

Dans cette partie, nous allons étudier les problèmes d'optimisation évoqués précédemment dans le cas où $H = \mathbb{R}^n$ muni du produit scalaire usuel et lorsqu'il n'y a pas de contraintes: nous nous intéressons donc à la minimisation de la fonction f sur tout l'espace. Ainsi nous considérons le problème formulé de la façon suivante:

$$(P) \quad \begin{cases} \min f(x) \\ x \in \mathbb{R}^n \end{cases} \quad \text{où } f \text{ est une fonction } \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$$

On admettra pour ce problème les résultats suivants dont les démonstrations (omisées ici) sont faciles à trouver en littérature.

Théorème 2.1: (Existence)

Soit f est une fonction $\mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ propre, continue et coercive. Alors le problème (P) admet au moins une solution.

Théorème 2.2: (Unicité)

Soit f est une fonction $\mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ strictement convexe. Alors le problème (P) admet au plus une solution.

Théorème 2.3: (Existence & Unicité)

Soit f une fonction de classe \mathcal{C}^1 de \mathbb{R}^n dans \mathbb{R} . on suppose qu'il existe un $\alpha > 0$ tel que

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \quad (\nabla f(x) - \nabla f(y))^t(x - y) \geq \alpha \|x - y\|^2$$

Alors f est une fonction strictement convexe et coercive: en particulier le problème (P) admet une solution unique.

Définition 2.1 (fonction elliptique)

Une fonction $f: H \rightarrow \mathbb{R}$ est elliptique si

$$\exists \alpha > 0 \text{ tel que } \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \quad (\nabla f(x) - \nabla f(y))^t(x - y) \geq \alpha \|x - y\|^2$$

Proposition:

Une fonction $f: H \rightarrow \mathbb{R}$ deux fois différentiable sur H est elliptique si et seulement si

$$\forall (x, y) \in H \times H \quad y^t D^2 f(x) y \geq \alpha \|y\|^2$$

Si on se restreint dans cet exposé aux cas de fonctions différentiables, plus précisément Gâteaux-différentiables; on pourra chercher des conditions qui portent sur la dérivée de la fonction à minimiser pour pouvoir calculer la ou les solutions en montrant que ces solutions proviennent de la résolution de certaines équations, ce qui facilite leurs calculs.

Conditions d'optimalité:

Théorème 2.4 (condition nécessaire d'optimalité du premier ordre)

Soit $f: H \rightarrow \mathbb{R}$ une fonction Gâteaux-différentiable sur H où H est un espace de Hilbert réel. Si x^* réalise un minimum (global ou local) de f sur H alors

$$\nabla f(x^*) = 0$$

Là c'est une condition nécessaire **c.à.d.** Pour que x^* puisse réaliser un minimum (global ou local) de f sur H , **il faut que** $\nabla f(x^*) = 0$.

*Note: Ce théorème n'a de sens que si la fonction f est différentiable en x^**

Note: Tout point x^* de H qui vérifie $\nabla f(x^*) = 0$ est appelé point critique ou point stationnaire. La relation $\nabla f(x^*) = 0$ est aussi appelé équation d'Euler.

Dans le cas où la fonction f est convexe, la condition du théorème précédent devient nécessaire et suffisante (CNS).

Théorème 2.5 (condition (CNS) d'optimalité du premier ordre)

Soit $f : H \rightarrow \mathbb{R}$ une fonction Gâteaux-différentiable et convexe sur H où H est un espace de Hilbert réel. x^* réalise un minimum (global ou local) de f sur H **si et seulement si**

$$\nabla f(x^*) = 0$$

Là c'est une condition nécessaire et suffisante c.à.d

- Si x^* réalise un minimum (global ou local) de f sur $H \Rightarrow \nabla f(x^*) = 0$
- Si $\nabla f(x^*) = 0 \Rightarrow x^*$ réalise un minimum (global ou local) de f sur H

Remarque:

Le cas où f est convexe est fréquent en pratique mais pas systématique. Cependant, on peut toujours raffiner le résultat précédent en considérant seulement la convexité locale de f au voisinage de x^* pour la réalisation de minimum local de f en x^* ou passer à des conditions qui font intervenir les dérivées secondes de f .

Théorème 2.6 (condition nécessaire d'optimalité du second ordre)

On suppose que x^* réalise un minimum local de f et que f est deux fois dérivable sur H **alors**

- $\nabla f(x^*) = 0$
- $\forall x \in H \quad x^t D^2 f(x^*) x \geq 0$

Dans le cas où $H = \mathbb{R}^n$, la deuxième condition est équivalent à dire que $D^2 f(x^*)$ qui représente la matrice Hessienne de f en x^* est semi-définie positive.

Là aussi les deux conditions sont **seulement nécessaires** car la réciproque du théorème est fausse.

Théorème 2.7 (condition suffisante du second ordre)

Soit une fonction f deux fois dérivable sur H vérifiant $\nabla f(x^*) = 0$ **et**

$$\exists \alpha > 0 \text{ tel que } \forall x \in H \quad x^t D^2 f(x^*) x \geq \alpha \|x\|^2$$

Alors la fonction f admet un minimum local strict en x^*

Dans le cas où $H = \mathbb{R}^n$, cette condition est équivalent à dire que $D^2 f(x^*)$ qui représente la matrice Hessienne de f en x^* est définie positive (donc strictement) c'est une condition d'ellipticité locale. Un choix possible de la valeur de α est celle de la plus petite valeur propre de la matrice $D^2 f(x^*)$.

Exemple 1:

Soit $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ définie par

$$f(x, y) = 3x^4 - 4x^2y + y^2$$

Il est facile de vérifier que $(0,0)$ est un point critique mais puisque $D^2 f(0,0) = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$ est seulement semi-définie positive car ses valeurs propres sont 0 et 2, alors on ne peut rien conclure. On peut toutefois remarquer qu'en ce point critique, il n'y a ni minimum ni maximum car $f(0,0) < f(0,1)$ et $f(0,0) > f(0.5,0.5)$.

Exemple de cas quadratique:

Prenons le cas où $f: \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction quadratique donnée par:

$$f(x) = x^t Ax - b^t x + c \quad \text{avec } A \text{ matrice carrée symétrique semi-définie positive, } c \text{ une constante et } b \in \mathbb{R}^n$$

Ainsi l'application quadratique f est un polynôme de degré 2 donc infiniment différentiable:

$$f(x) = f(x_1, x_2, \dots, x_n) = \underbrace{\frac{1}{2} \sum_{i=1}^n a_{ii} x_i^2 + 2 \sum_{i<j} a_{ij} x_i x_j}_{\text{forme quadratique}} - \underbrace{\sum_{i=1}^n b_i x_i}_{\text{forme linéaire}} + \underbrace{c}_{\text{constante}}$$

en posant

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n; A^t = A = (a_{ij})_{\substack{i=1,n \\ j=1,n}} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \in M_n(\mathbb{R}); b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^n$$

les points critiques de f sont données par les solutions de $\nabla f(x) = Ax - b = 0$ et sont donc les solutions du système $Ax = b$, mais puisque $\nabla^2 f(x) = A$ pour tout x , alors $f(x)$ possède automatiquement un minimum global en ces points critiques.

Contrairement au cas d'application quelconque, la convexité d'une application quadratique est totalement caractérisée par sa matrice Hessienne.

Théorème 2.8

Soit f une application quadratique sur \mathbb{R}^n , $f(x) = x^t Ax - b^t x + c$ **Alors**

- f convexe $\Leftrightarrow A$ semi-définie positive
- f strictement convexe $\Leftrightarrow f$ fortement convexe $\Leftrightarrow A$ définie positive

Théorème 2.9

Pour une application quadratique sur \mathbb{R}^n , $f(x) = x^t Ax - b^t x + c$ et $Ax^* = b$ pour $x^* \in \mathbb{R}^n$.

- Si A est semi-définie positive (resp. négative), **alors** x^* est un minimum (resp. maximum) global de f et x^* est une solution du système d'équations linéaires $Ax = b$.
- Si A est définie positive (resp. négative), **alors** f admet un **unique** minimum (resp. maximum) global.
- Si A **n'est pas** semi-définie positive (resp. négative), **alors** f n'admet **aucun** minimum (resp. maximum) global ou local.

Exemple numérique de cas quadratique:

Soit l'application quadratique suivante:

$$f(x) = \frac{1}{2} x^t Ax - b^t x + c, \quad \text{avec}$$

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n; A^t = A = (a_{ij})_{\substack{i=1,n \\ j=1,n}} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \in M_n(\mathbb{R}); b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^n$$

1. Exprimer $\frac{\partial f}{\partial x_i}(x)$, en déduire $\nabla f(x)$; Exprimer $\frac{\partial^2 f}{\partial x_i \partial x_j}(x)$, en déduire $\nabla^2 f(x)$.

2. Pour quelle condition $f(x)$ est-elle convexe ou strictement convexe? quel extrémum aura-t-elle alors pour cette condition?

3. Pour $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$; $b = \begin{pmatrix} -3 \\ 1 \\ -2 \end{pmatrix}$; $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$;

- Déterminer l'extrémum de $f(x)$ ---Est - il unique? Pourquoi?
- Montrer qu'il existe un $\alpha > 0$ tel que $A \geq \alpha I$, avec I une matrice identité.
- En déduire la solution du système d'équations suivant:

$$\begin{cases} 2x - y = -3 \\ -x + 2y - z = 1 \\ -y + 2z = -2 \end{cases}$$

Solution:

1) $\frac{\partial f}{\partial x_i}(x) = \sum_{j=1}^n a_{ij} x_j - b_i$; **donc** $\nabla f(x) = Ax - b$; $\frac{\partial^2 f}{\partial x_i \partial x_j}(x) = a_{ij}$; **donc** $\nabla^2 f(x) = Hf(x) = A$.

2) f est convexe si A est semi-définie positive et aura un minimum local; f est strictement convexe si A est définie positive et dans ce cas elle aura un minimum global (unique).

3)

a) Puisque $Hf(x)=A$; d'après le polynôme caractéristique les 3 valeurs propres sont >0 alors

A est définie positive et $f(x)$ possède donc un minimum global en $x^* = \begin{pmatrix} -9/4 \\ -3/2 \\ -7/4 \end{pmatrix}$. Ce

minimum est unique car $f(x)$ est strictement convexe.

b) il existe un $\alpha > 0$ tel que $A \geq \alpha I$ puisque A est définie positive et ses 3 valeurs propres $\lambda_1 < \lambda_2 < \lambda_3$ alors puisque $A \geq \lambda_1 I$, α peut être au moins égale à λ_1 .

c) Le système proposé est exactement $Ax=b$ avec $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$ et $b = \begin{pmatrix} -3 \\ 1 \\ -2 \end{pmatrix}$

La solution de ce système est celle qui annule $\nabla f(x) = Ax - b$ autrement dit $x^* =$

$$\begin{pmatrix} -9/4 \\ -3/2 \\ -7/4 \end{pmatrix}.$$

Remarque générale:

Pour une fonction $f: D \rightarrow \mathbb{R}$, le minimum s'il existe, est unique. Cependant, il peut être atteint en plusieurs points différents. En effet, le minimum d'une fonction s'il existe est nécessairement la borne inférieure de l'ensemble des valeurs prises par la fonction f sur l'ensemble D . il doit y exister donc une suite $\{x_n\}$ de points de D telle que la suite $\{f(x_n)\}$ converge vers ce minimum. La suite $\{x_n\}$ est appelée suite minimisante. La question de l'existence de ce minimum serait liée à la convergence de cette suite vers une certaine limite et à l'appartenance de cette dernière au domaine D .

Résolution Algorithmique:

D'une manière générale, la résolution algorithmique du problème d'optimisation (P) consiste à se rapprocher de l'optimum (minimum dans notre cas) en construisant une suite $\{x_n\}_{n \in \mathbb{N}}$ de \mathbb{R}^n qui converge vers une solution x^* . Cela en considérant des conditions suffisantes sur f de façon à ce que $x_n \rightarrow x^*$.

Plusieurs méthodes s'offrent à nous pour la résolution de notre problème d'optimisation sans contraintes (P):

- 1) Ils existent des méthodes directes n'utilisant pas les dérivées de la fonction objective f telle que la méthode Hook. Ces méthodes sont utiles pour les applications non différentiables mais leurs convergences sont généralement lentes.
- 2) Les méthodes de descentes qui utilisent les dérivées premières de la fonction objective f , leur principe consiste à construire l'itéré x_{k+1} en choisissant une direction de descente d_k et un pas ρ_k tel que $x_{k+1} = x_k + \rho_k d_k$. c'est le cas de la méthode de relaxation et celle du gradient avec toutes ses variantes.
- 3) Les méthodes qui utilisent les dérivées secondes de la fonction objective f , il s'agit essentiellement de la méthode de Newton qui n'est pas une méthode d'optimisation mais plutôt une méthode d'approximation des zéros d'une fonction. Son utilisation en optimisation consiste donc à l'appliquer sur le $\nabla f(x)$. Sa convergence est la plus rapide à condition de choisir le point initial proche du minimum qu'on cherche puisque c'est une méthode de recherche locale.

Notez bien qu'en optimisation la convergence, la rapidité de la convergence, la stabilité et la complexité du calcul des itérations sont des critères essentielles pour tous ces algorithmes.

Définition:

Un algorithme itératif est défini par une application vectorielle $\mathbb{R}^n \rightarrow \mathbb{R}^n$ qui génère une suite de vecteurs x_k par construction généralement récurrente et en introduisant une erreur vectorielle $e_k = x^* - x_k$.

- ❖ Si $\exists \alpha < 1, k_0 \in \mathbb{N}$ et $k > k_0$ tel que $\|e_{k+1}\| \leq \alpha \|e_k\|$, on dit alors que la convergence est géométrique.
- ❖ Si $\exists k_0 \in \mathbb{N}$ et $k > k_0$ tel que $\|e_{k+1}\| \leq \|e_k\|$, on dit alors que la convergence est linéaire.
- ❖ Si $\exists k_0 \in \mathbb{N}$ et $k > k_0$ tel que $\|e_{k+1}\| \leq \|e_k\|^p, p \in \mathbb{N}^*, p \geq 2$, la convergence est dite quadratique si $p=2$.
- ❖ Si à partir de $k > k_0$, on a $\|e_k\| = 0$, la convergence est dite finie.

En pratique, dans toutes les méthodes itératives, on peut accepter x_k comme approximation de la solution dès que $\|x_k - x_{k-1}\| \leq tol$ c.à.d. en se fixant tol comme un critère d'arrêt.

Technique itérative de base:

Commençant par une méthode itérative de base pour le calcul de point fixe d'une fonction. En pratique, on est souvent confronté à la résolution d'un système d'équations non linéaires de la manière suivante:

Soit une fonction $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ donnée, on cherche un point $x \in \mathbb{R}^n$ tel que

$$f(x) = 0$$

En général, il n'y a pas d'algorithme fini pour résoudre ce problème, on est donc obligé d'utiliser des méthodes itératives. D'autant plus, sans hypothèses supplémentaires, on ne sait rien sur l'existence de solution, par exemple si $f(x) = e^x$ il n'y a pas de solution, pour $f(x) = \sin(x)$ il y en a une infinité. Voyons plutôt avec la méthode ci-dessous.

Méthode des approximations successives:

On considère le problème du calcul du point fixe de l'application $\varphi: \mathbb{R}^n \rightarrow \mathbb{R}^n$ c.à.d., on cherche $x \in \mathbb{R}^n$ tel que $x = \varphi(x)$

Ce deuxième problème est équivalent au premier problème $f(x) = 0$ (les deux problèmes possèdent les mêmes solutions) et il y a plusieurs possibilités pour se ramener au premier problème. Par exemple, on peut définir φ comme $\varphi = x - f(x)$ ou bien $\varphi = x - Bf(x)$ avec B un nombre non nul ou une matrice bien choisie puis on se donne une approximation x_0 (arbitraire) et on considère la méthode itérative suivante:

$$x_{k+1} = \varphi(x_k)$$

Si la suite $\{x_k\}$ converge, disons vers $a \in \mathbb{R}^n$, et si la fonction $\varphi(x)$ est continue en a , la limite a est une solution $x = \varphi(x)$.

D'une manière générale, si $\varphi(x)$ est 2 fois continûment différentiable et si $a \in \mathbb{R}^n$ est une solution de $x = \varphi(x)$, l'erreur $e_k = x_k - a$ satisfait

$$e_{k+1} = \varphi'(a) e_k + O(\|e_k\|^2)$$

En effet, comme $a = \varphi(a)$, on a

$$e_{k+1} = x_{k+1} - a = \varphi(x_k) - \varphi(a) = \varphi'(a) e_k + O(\|e_k\|^2)$$

Plusieurs conclusions peuvent être tirées de ce résultat:

- Si $\varphi'(a)$ possède une valeur propre $|\lambda_i| > 1$, la composante e_k dans la direction du vecteur propre v_i va être agrandie \rightarrow la méthode ne converge pas vers a .
- Si toutes les valeurs propres de $\varphi'(a)$ satisfont $|\lambda_i| < 1$; on peut choisir une norme dans \mathbb{R}^n telle que $\|\varphi'(a)\| < 1$ ceci implique que pour $\|e_k\|$ suffisamment faible, on a $\|e_{k+1}\| \leq \alpha \|e_k\|$ où α est un nombre entre $\varphi'(a)$ et 1 donc l'erreur e_k converge vers zéro.

Méthodes du gradient:

Toujours pour le cas sans contraintes, la méthode du gradient fait partie d'une grande classe de méthodes numériques appelée méthodes de descente dont le principe est:

En se donnant un point de départ arbitraire x_0 , pour se rapprocher du minimum d'une fonction $f(x)$, la suite $\{x_k\}$ est construite de façon que $f(x_{k+1}) < f(x_k)$ en espérant que x_k converge vers l'optimum recherché lorsque k tend vers l'infini i.e. $x_k \rightarrow x^*$ quand $k \rightarrow +\infty$ puis par continuité de la fonction $f(x)$ nous pouvons conclure que: $f(x_k) \rightarrow f(x^*) = \min f(x)$.

La suite récurrente utilisée dans les méthodes de descente est de la forme:

$$x_{k+1} = x_k + \rho_k d_k$$

En choisissant une direction de descente d_k et un pas ρ_k , on peut progresser dans les itérations.

Le problème est que pour un certain x_k , comment choisir la direction de descente d_k pour que $f(x_{k+1})$ soit toujours inférieur ou égale à $f(x_k)$? Cela peut être réalisé en assurant la maximisation de la quantité $(f(x_k) - f(x_{k+1}))$ dans le développement de Taylor d'ordre 1.

En effet à partir de

$$f(x_{k+1}) = f(x_k + \rho_k d_k) \cong f(x_k) + \rho_k \nabla f(x_k)^t d_k + O^2(\rho_k, d_k)$$

En négligeant les termes d'ordre supérieurs, on remarque que

Maximiser $(f(x_k) - f(x_{k+1}))$ est équivalent à maximiser $-(\rho_k \nabla f(x_k)^t d_k)$

Or pour $\rho_k > 0$, nous avons

$$\|-\rho_k \nabla f(x_k)^t d_k\| \leq \rho_k \|\nabla f(x_k)^t\| \|d_k\|$$

On peut donc choisir $d_k = -\nabla f(x_k)$

Et on aura

$$x_{k+1} = x_k - \rho_k \nabla f(x_k)$$

Notez bien que la direction du vecteur gradient $\nabla f(x_k)$ est celle où la fonction augmente, donc la direction de $-\nabla f(x_k)$ est effectivement une direction de descente.

Algorithme:

1. Initialisation, $k=0$, choix de x_0 et ρ_0
2. Itération : $x_{k+1} = x_k - \rho_k \nabla f(x_k)$
3. Arrêt : si $\|e_k\| < \varepsilon$ stop
Sinon $k=k+1$ et revenir à 2.

Le plus grand avantage de la méthode du gradient est sa facilité de mise en œuvre. Malheureusement, les conditions de convergence sont assez lourdes et aussi la vitesse de convergence est généralement lente.

Les conditions de convergence sont essentiellement la convexité stricte de la fonction objective f et la Lipchicité de son gradient, les deux conditions sont les suivantes:

- 1) $\exists \alpha > 0$ tel que $\forall x, y \in \mathbb{R}^n$ $(x - y)^t (\nabla f(x) - \nabla f(y)) \geq \alpha \|x - y\|^2$
C'est la propriété de la α -convexité
- 2) $\exists M > 0$ tel que $\forall x, y \in \mathbb{R}^n$ $\|(\nabla f(x) - \nabla f(y))\| \leq M \|x - y\|$
C'est-à-dire $\nabla f(x)$ Lipchitz

Théorème 2.10

Si la fonction $f(x)$ vérifie ces deux conditions et le pas est choisi suffisamment proche de zéro, **alors** l'algorithme du gradient converge vers $\min f(x)$.

Théorème 2.11

Si la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est de classe \mathcal{C}^1 , coercive et strictement convexe et de $\nabla f(x)$ Lipchitz de constante M . **Alors** l'algorithme du gradient converge vers $\min f(x)$ si ρ_k est choisi dans l'intervalle $[a, b]$ avec $b < \frac{2}{M}$ ou $(\frac{2\alpha}{M})$ avec α le coefficient de la α -convexité).

Remarque:

En général, il est difficile de calculer α et M (en supposant qu'ils existent).

Alors dans la pratique on prend un pas assez petit pour être sûr d'avoir la convergence. Mais dans ce cas la convergence peut être très lente. Un exemple où on peut calculer α et M : Si la fonction f est quadratique avec une matrice A qui est SDP. Alors on peut prendre $\alpha = \lambda_{\min} > 0$ (la plus petite valeur propre de A) et $M = \|A\|_2 = \rho(A) = \lambda_{\max} > 0$ (la plus grande valeur propre de A).

Sélection d'un pas ρ_k adéquat

Le choix est décisif pour la sélection de la direction de descente d_k , nous avons pu déterminer exactement la direction de descente d_k convenable. Alors que le choix du pas ρ_k n'est pas toujours évident, en effet, le plus souvent on utilise pour la méthode du gradient un **pas constant** ($\rho_k = \text{constant}$). Toutefois, on peut faire varier le pas à chaque itération, et on obtient alors la méthode **du gradient à pas variable**. On peut aller plus loin et déterminer le pas optimal à chaque itération, cela en cherchant le pas qui rend la fonction objective minimale le long de la direction de descente et on obtient ce qu'on appelle la méthode du **gradient à pas optimal**.

Gradient à pas optimal

En effet, si nous considérons : $q_k(\rho_k) = f(x_k + \rho_k d_k)$, nous allons avoir une valeur optimale de ρ_k : $\rho_{kopt} = \operatorname{argmin}_{\rho_k} f(x_k + \rho_k d_k)$.

À titre d'exercice, nous allons voir que pour une fonction quadratique $f(x) = \frac{1}{2}x^tAx - b^tx + c$

Nous obtenons: $\rho_{kopt} = \frac{\|Ax_k - b\|^2}{(Ax_k - b)^t A (Ax_k - b)}$

D'une manière générale, pour un $\rho_k > 0$, nous avons par application de la règle de la dérivation en chaîne:

$$q_k'(\rho_k) = \nabla f(x_k + \rho_k d_k) \cdot d_k$$

Puisque d_k est une direction de descente, on a $q_k'(0) = \nabla f(x_k) \cdot d_k < 0$, il s'en suit que si on prend un ρ_k très proche de zéro, on n'est sûr de faire décroître q_k . Toutefois, il faut faire attention car:

- Si ρ_k trop grand, on risque de ne pas faire décroître la fonction q ou obtenir un comportement oscillant.
- Si ρ_k trop petit, l'algorithme n'avancera pas très vite

En pratique, on ne peut chercher à chaque fois le minimum $q_k(\rho_k)$, plutôt, à chaque itération on détermine le ρ_k en effectuant une recherche linéaire selon l'une des deux méthodes classiques de Wolfe ou de Goldstein.

Algorithme de Wolfe:

- 1) **Initialisation:** choisir m_1 et m_2 tel que $0 < m_1 < m_2 < 1$. Poser $\rho = 1, \rho_+ = \rho_- = 0$
- 2) **Si** $q_k(\rho) \leq q_k(0) + m_1 \rho q_k'(0)$ et $q_k'(\rho) \geq m_2 q_k'(0)$, **Stop** $\rho_k = \rho$

Sinon

- **Si** $q_k(\rho) > q_k(0) + m_1 \rho q_k'(0)$, on pose $\rho_+ = \rho$
- **Sinon** si $q_k(\rho) \leq q_k(0) + m_1 \rho q_k'(0)$ et $q_k'(\rho) < m_2 q_k'(0)$, on pose $\rho_- = \rho$
- **Fin si**

Fin si

- 3) **Si** $\rho_+ = 0$, alors choisir $\rho > \rho_-$ (par exemple $\rho = 2\rho_-$) et aller à 2

Sinon si $\rho_+ > 0$, alors choisir $\rho \in]\rho_-, \rho_+[$ (par exemple $\rho = \frac{\rho_- + \rho_+}{2}$) et aller à 2

Fin si

L'algorithme associé à la règle de Goldstein étant similaire à celui de Wolfe.

Remarque:

- En général, l'algorithme trouve un point critique qui est un minimum local.
- L'algorithme peut ne pas converger (aller vers $-\infty$).
- S'il y a plusieurs minima locaux et que l'algorithme converge, on ne peut savoir vers quel minima va converger.
- Pour un problème de maximisation, on choisit $d_k = \nabla f(x_k)$ et on maximise pour la recherche linéaire.
- L'algorithme s'approche du point critique en zigzaguant.

Méthodes du gradient conjugué:

Malgré que le vecteur gradient d'une fonction représente la direction dans laquelle la fonction a la plus forte augmentation et donc l'opposé du gradient est la plus grande direction de descente locale, l'utilisation de la méthode du gradient ne nous permet pas de converger le plus rapidement possible vers le minimum.

En s'inspirant de la méthode du gradient, nous allons introduire la méthode du gradient conjugué qui est une méthode plus exacte et particulièrement rapide, et converge en au plus n itérations lorsque la fonction est quadratique avec une matrice (A d'ordre n) symétrique définie positive. L'idée est de construire la suite $\{x_k\}$ avec des directions de descente d_k orthogonales (ou conjuguées) pour le produit scalaire associé à la matrice A défini par: à tout couple de vecteurs (x,y) de \mathbb{R}^n , on associe la quantité x^tAx .

Pour cela, et lorsque la fonction est quadratique, les directions de descente w_k ne sont plus égales à $d_k = Ax_k - b$ mais corrigées de façon à construire des directions w_k orthogonales (ou conjuguées) pour le produit scalaire associé à la matrice A . Plus exactement:

On pose:

$$w_k = d_k + \alpha_k w_{k-1} \quad \text{où } \alpha_k \text{ est tel que } w_k^t A w_{k-1} = 0$$

Algorithme de la méthode du gradient conjugué:

(Pour une fonction quadratique $f(x) = \frac{1}{2}x^tAx - b^tx + c$)

- **Initialisation** : ($k=0$)

choisir un nombre d'itération maximum k_{\max} puis choix de $x_0 \in \mathbb{R}^n$; $\varepsilon_0 > 0$; $\varepsilon_1 > 0$; $d_0 = \nabla f(x_0) = Ax_0 - b$ et calculer $\rho_0 = -\frac{\|d_0\|^2}{d_0^t A d_0}$; $w_0 = d_0$; $x_1 = x_0 + \rho_0 d_0$

- **Itération k** : ($k \geq 1$)

1. Si ($\|x_{k+1} - x_k\| < \varepsilon$) ou ($k > k_{\max}$) ou ($\|d_k\| < \varepsilon_1$) alors **Stop**

2. $\alpha_k = -\frac{d_k^t A w_{k-1}}{w_{k-1}^t A w_{k-1}}$; $w_k = d_k + \alpha_k w_{k-1}$ puis on calcule

$$\rho_k = -\frac{d_k^t w_k}{w_k^t A w_k}$$

$$x_{k+1} = x_k + \rho_k w_k$$

$$d_{k+1} = Ax_{k+1} - b$$

- **$k = k + 1$** et on retourne à l'itération k

La méthode du gradient conjugué est très stable même pour une matrice A mal conditionnée. Dans le cas de matrice pleine d'ordre n , elle demande $2n^3$ opérations et n itérations. Le nombre d'opérations diminue nettement lorsque la matrice est creuse.

Méthode de Newton:

D'une part, on sait que pour rechercher l'extremum d'une fonction différentiable $f(x)$, on peut se ramener d'abord à rechercher ses point critiques donnés par $\nabla f(x) = 0$. D'autre part, la méthode de Newton n'est pas une méthode d'optimisation mais plutôt une méthode de recherche de zéro d'une fonction selon la forme $f(x)=0$ dont la résolution n'est pas toujours possible et une solution approchée est à considérer. L'idée de l'utilisation de la méthode de Newton en optimisation est de l'appliquer à résoudre l'équation $\nabla f(x) = 0$ qui se réduit à la résolution de système de $n \times n$ équations pour une fonction $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Voyons d'abord le cas de fonction scalaire mono-variable $f: \mathbb{R} \rightarrow \mathbb{R}$.

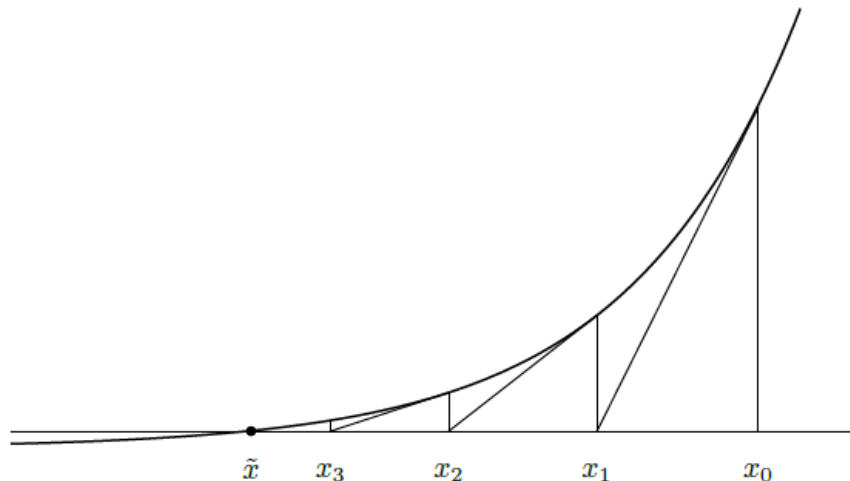
Soit une fonction dérivable $f: \mathbb{R} \rightarrow \mathbb{R}$, on cherche x^* tel que $f(x^*)=0$; on se donne un x_0 comme approximation de la solution or au voisinage de cette solution x_0 , on a:

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \approx 0$$

si $f'(x_0) \neq 0$, alors $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$ et on construit la suite récurrente $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$

Algorithme: (cas monovariante)

- Choisir une approximation initiale x_0 de la solution x^*
- **Pour** $k=0, 1, 2, \dots$
 - calculer $f(x_k)$ et $f'(x_k)$
 - calculer $\Delta x_k = -\frac{f(x_k)}{f'(x_k)}$
 - $x_{k+1} = x_k + \Delta x_k$
- **Fin pour** (qu'il faut compléter par un critère d'arrêt.)



La méthode de descente de Newton pour la recherche du zéro d'une application $f: \mathbb{R} \rightarrow \mathbb{R}$

Pour une fonction différentiable $f: \mathbb{R}^n \rightarrow \mathbb{R}$,

La minimisation de $f(x)$ passe par la résolution de $\nabla f(x) = F(x) = 0$, si de plus $\nabla f(x) = F(x)$ est de classe $\mathcal{C}^1 \mathbb{R}^n \rightarrow \mathbb{R}^n$, on aura pour résoudre $F(x) = 0$ avec $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$:

$$\begin{cases} x_0 \in \mathbb{R}^n \\ x_{k+1} = x_k - \frac{F(x_k)}{\nabla F(x_k)} \end{cases} \equiv \begin{cases} x_0 \in \mathbb{R}^n \\ x_{k+1} = x_k - (D^2 f(x_k))^{-1} \nabla f(x_k) \end{cases}$$

Dans la méthode de Newton, on utilise indifféremment les notations suivantes: $\nabla F(x_k)$, $D^2 f(x_k)$ et $DF(x(k))$ pour " $Hf(x_k)$ " selon le contexte. De même pour $\nabla f(x)$ et $F(x)$.

L'algorithme de la méthode appliquée à $F(x)$ est alors le suivant:

Algorithme: (cas multidimensionnelle.)

- Pour $k=0$.
- Choisir une approximation $x_0 \in \mathbb{R}^n$ (au voisinage de x^*) initiale de la solution.

- Choisir $\varepsilon > 0$
- Tant que $(x_{k+1} - x_k) \geq \varepsilon$ et $(k \leq k_{\max})$ faire
 - Résoudre le système linéaire $DF(x_k)\Delta x_k = -F(x_k)$ (pour trouver Δx_k)
 - Poser $x_{k+1} = x_k + \Delta x_k$
 - $k=k+1$
- Fin tant que

Convergence de la méthode de Newton:

Pour étudier la convergence de la méthode de Newton en optimisation, l'application se fait sur le $\nabla f(x) = F(x)$, si $f(x)$ est trois fois différentiable et en notant $DF(x) = F'(x)$ et $D^2F(x) = F''(x)$, nous considérons le développement en série de Taylor de la fonction $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

$$F(x) \approx F(x_k) + F'(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^t F''(x_k)(x - x_k) + O(\|x - x_k\|^3)$$

Si l'on pose $x = x^*$ avec x^* solution du développement au premier ordre, on a donc

$$0 = F(x_k) + F'(x_k)(x_{k+1} - x_k)$$

Ainsi, en notant $e_k = x_k - x^*$, on obtient si x^* est suffisamment proche de x_k :

$$0 = F'(x_k)(-e_{k+1}) + \frac{1}{2} e_k^t F''(x_k) e_k + O(\|e_k\|^3)$$

Enfin, on a:

$$e_{k+1} = \frac{1}{2} (F'(x_k))^{-1} e_k^t F''(x_k) e_k + O(\|e_k\|^3)$$

Ce qui montre que la convergence de cette méthode est quadratique c.à.d. que si on choisit x_0 très proche de la solution x^* (qu'on cherche justement) alors la suite $\{x_k\}$ converge vers x^* de façon quadratique (donc rapidement).

Remarque:

On peut facilement prouver que si $\nabla f(x^*) = 0$ et $Hf(x^*)$ est inversible alors il existe $\varepsilon > 0$ tel que si $x_0 \in B(x^*, \varepsilon)$ alors la suite $\{x_k\}$ est bien définie $x_k \rightarrow x^*$ (quadratiquement) lorsque $k \rightarrow +\infty$.

Même si on a pu confirmer que la convergence n'est que locale, l'avantage de cette méthode est qu'elle est générale et c'est *la seule qui ne nécessite pas la convexité* de la fonction de coût.

Méthode de Newton pour une fonction convexe:

Si la fonction objective est convexe, on peut facilement interpréter la méthode de Newton comme une méthode de descente.

En effet, on peut écrire l'expression

$$x_{k+1} = x_k - (Hf(x_k))^{-1} \nabla f(x_k) \quad \text{sous forme} \quad x_{k+1} = x_k + \rho_k d_k$$

Avec

$$\rho_k = 1 \text{ et } d_k = -(Hf(x_k))^{-1} \nabla f(x_k)$$

Évidemment, pour une fonction convexe $Hf(x_k)$ est une matrice symétrique définie positive puisqu'elle est supposée inversible. Ce qui conduit à:

$$-d_k^t \nabla f(x_k) > 0$$

Ce qui prouve que d_k est une direction de descente.

Note d'implémentation numérique:

Si la facilité de mise en œuvre et la rapidité de la convergence sont les atouts de l'utilisation de la méthode de Newton, elle nécessite en revanche le calcul du gradient et de la matrice Hessienne (dont les formes analytiques sont souvent inconnues) qui dès fois s'avèrent coûteux ou même impossible. Une première idée consiste à s'aider de l'approximation des éléments $\partial f_i / \partial x_j$ de la Jacobienne comme suit:

$$\frac{\partial f_i}{\partial x_j}(x_1, \dots, x_n) \approx \frac{f_i(x_1, \dots, x_j + \delta, \dots, x_n) - f_i(x_1, \dots, x_n)}{\delta}$$

En choisissant $\delta \cong \sqrt{\text{eps}}$ où *eps* est la précision de la machine utilisée. Toutefois, *Il existe aussi d'autres variantes de la méthode de Newton qui évitent le calcul de l'inverse de la Hessienne.*

Si l'avantage de la méthode de Newton est qu'elle ne nécessite pas la convexité de la fonction objective et possède une grande rapidité de convergence (convergence quadratique), un des gros problèmes de cette méthode est que le choix de x_0 joue un grand rôle sur la convergence ou non de la méthode de Newton: la méthode est **très sensible** à l'initialisation. Rappelons les inconvénients suivants:

- la méthode peut diverger si le point de départ est trop éloigné de la solution,
- la méthode n'est pas définie si la matrice Hessienne n'est pas inversible,
- la méthode peut converger indifféremment vers un minimum, un maximum ou un point de selle.

Une approche possible consiste à faire tourner quelques itérations d'une méthode de gradient pour approcher x^* et de considérer l'itéré résultant comme le point de départ de la méthode de Newton.

Si nous revenons à l'inconvénient du calcul et l'inversion de la matrice $DF(x(k))$ à chaque itération. Certaines méthodes proposent (surtout pour des problèmes de grandes dimensions) d'utiliser non pas $DF(x(k))$ comme matrice mais plutôt une approximation de celle-ci (plus précisément, dans les méthodes de **quasi-Newton** décrites ci-dessous, on construit une suite de matrices $S(k)$ qui approchent $[DF(x(k))]^{-1}$.

Méthodes quasi-Newton

L'idée ici est de reprendre l'algorithme de Newton tout en évitant de calculer $DF(x(k))$ et son inverse. Pour cela, à l'itération k , nous allons chercher à construire une approximation $S(k)$ symétrique définie positive de $[DF(x(k))]^{-1}$ et $\rho(k)$ un paramètre positif fourni par un algorithme de minimisation unidimensionnel le long de la direction $d(k) = -S(k) \nabla f(x_k)$ tels que l'opération classique.

$$x_{k+1} = x_k - [D^2 f(x(k))]^{-1} \nabla f(x_k)$$

Soit remplacée par l'opération simple et moins coûteuse suivante:

$$x_{k+1} = x_k - \rho_k S_k \nabla f(x_k)$$

L'objectif est de trouver une bonne approximation S_k de $[D^2 f(x(k))]^{-1}$ en fonction de S_{k-1} , $\nabla f(x_k)$ et $\nabla f(x_{k-1})$ c.à.d. telle que la différence soit la plus petite possible selon une certaine norme matricielle. En général, on utilisera l'une des deux méthodes de calcul suivantes:

- La formule de Davidon-Fletcher-Powell (DFP)

$$S_k = S_{k-1} - \frac{S_{k-1} Y_k Y_k^t S_{k-1}}{Y_k^t S_{k-1} Y_k} + \frac{D_{k-1} D_{k-1}^t}{D_{k-1} Y_k}$$

- La formule de Broyden-Fletcher-Goldfarb-Shanno (BFGS)

$$S_k = S_{k-1} + \left(1 + \frac{Y_k^t S_{k-1} Y_k}{Y_k^t D_{k-1}}\right) \frac{D_{k-1} D_{k-1}^t}{Y_k^t D_{k-1}} - \frac{S_{k-1} Y_k D_{k-1}^t + D_{k-1} Y_k^t S_{k-1}}{Y_k^t D_{k-1}}$$

Avec

$$Y_k = \nabla f(x_k) - \nabla f(x_{k-1})$$

Cette dernière méthode est considérée comme très robuste et est moins sensible aux erreurs dans les recherches linéaires, Nous pourrions même citer un théorème de convergence dans le cas quadratique. La robustesse des précédents algorithmes est intimement liée à la recherche du pas optimal $\rho(k)$. la recherche en ligne de ce dernier consiste à utiliser la direction de Newton lorsqu'elle est définie sinon utiliser celle qui a la plus grande pente

En pratique, la direction de descente d est calculée en résolvant:

$$(DF(x_k) + \Delta)d = -F(x_k)$$

Où Δ est une matrice diagonale telle que $(DF(x_k) + \Delta)$ est définie positive pour assurer que d soit une direction de descente.

Dans ce cas, l'algorithme de Newton avec recherche en ligne aura la forme suivante:

Algorithme: l'algorithme de Newton avec recherche en ligne

- **Choisir** une approximation initiale x_0 de la solution x^*
- **Tant que** critère d'arrêt non satisfait, faire
 - calculer d solution de $(DF(x_k) + \Delta)d = -F(x_k)$ i.e. $((\nabla^2 f(x_k) + \Delta)d = -\nabla f(x_k))$
 - Trouver un pas ρ qui minimise $f(x_k + \rho_k d_k)$
 - $x_{k+1} = x_k + \rho_k d_k$
- **Fin tant que**

Cette forme de recherche en ligne est utilisée dans les deux algorithmes (DFP) et (BFGS).

Algorithme de Davidon-Fletcher-Powell (DFP)

- **Choisir** une approximation initiale $x_0 \in \mathbb{R}^n$ de la solution x^*
- **Poser** $k=0$
- **Choisir** S_0 matrice symétrique définie positive
- **Choisir** $\varepsilon > 0$
- **Tant que** $(x_{k+1} - x_k) \geq \varepsilon$ et $(k \leq k_{\max})$, faire
 - **Calculer** ρ_k par une recherche linéaire $d_k = -S_k \nabla f(x_k)$
 - **Poser** $x_{k+1} = x_k + \rho_k d_k$
 - **Poser** $D_{k+1} = x_{k+1} - x_k$
 - **Calculer** $Y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
 - **calculer** $S_{k+1} = S_k - \frac{S_k Y_k Y_k^t S_k}{Y_k^t S_k Y_k} + \frac{D_k D_k^t}{D_k Y_k}$

- **Fin tant que**

Algorithme de Broyden-Fletcher-Goldfarb-Shanno (BFGS)

- **Choisir** une approximation initiale $x_0 \in \mathbb{R}^n$ de la solution x^*
- **Poser** $k=0$
- **Choisir** S_0 matrice symétrique définie positive

- Choisir $\varepsilon > 0$
- Tant que $(x_{k+1} - x_k) \geq \varepsilon$ et $(k \leq k_{\max})$, faire
 - Calculer ρ_k par une recherche linéaire $d_k = -S_k \nabla f(x_k)$
 - Poser $x_{k+1} = x_k + \rho_k d_k$
 - Poser $D_{k+1} = x_{k+1} - x_k$
 - Calculer $Y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$

- calculer

$$S_{k+1} = S_k + \left(1 + \frac{Y_k^t S_k Y_k}{Y_k^t D_k}\right) \frac{D_k D_k^t}{Y_k^t D_k} - \frac{S_k Y_k D_k^t + D_k Y_k^t S_k}{Y_k^t D_k}$$

- Fin tant que

Chapitre 3

Optimisation Sous Contraintes

I. Minimisation avec (ou sous) contraintes:

Dans la plupart des problèmes d'optimisation, les variables ne sont pas libres de prendre n'importe quelle valeur, elles sont habituellement restreintes à un domaine. Il est aussi essentiel de définir au mieux les critères d'optimalité qui vont permettre de caractériser les solutions. Par exemple, dans le cas des robots manipulateurs deux critères sont couramment utilisés : le temps minimal et l'énergie minimale. En fait, le premier critère va générer des solutions mettant à dure épreuve les capacités mécaniques de la structure alors que le deuxième, qui d'ailleurs ne correspond pas vraiment à l'énergie minimale, va fournir des commandes souples en limitant les efforts sur les actionneurs. Une pondération de ces deux critères fournit des solutions de compromis de bonne qualité. L'optimalité dans cette optique est donc le moyen de permettre à l'utilisateur de choisir à un niveau supérieur, voire stratégique. Nous nous trouvons dans ce cas face à un véritable outil d'aide à la décision.

Dans cette section, nous nous intéressons au problème d'optimisation dans sa forme générale où on considère une fonction $f: D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ avec D un ouvert et le problème suivant:

$$(P2) \quad \text{Minimiser } f(x) \text{ tel que } x \in D$$

Où D ici désigne l'ensemble des variables x admissibles c.à.d. celles qui satisfont toutes les contraintes et représente ainsi le domaine de définition de la fonction de coût $f(x)$.

Nous traiterons plus particulièrement le cas où D est défini par des égalités et des inégalités :

$$D = \{x \in \mathbb{R}^n \mid g(x) \leq 0, \quad i = 1, 2, \dots, p \text{ et } h(x) = 0, \quad i = 1, 2, \dots, q\} \text{ avec } x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$h: \mathbb{R}^n \rightarrow \mathbb{R}^q$ représente q contraintes d'égalités $h(x) = (h_1(x), h_2(x), \dots, h_q(x))$; h est supposée continue.

$g: \mathbb{R}^n \rightarrow \mathbb{R}^p$ représente p contraintes d'inégalités $g(x) = (g_1(x), g_2(x), \dots, g_p(x))$; g est supposée continue.

Les contraintes d'égalités et d'inégalités sont à considérer élément par élément.

Rappelons qu'un ensemble D est fermé si tout point x de la frontière de D appartient à D , et un ensemble D est borné s'il existe une constante $M \in \mathbb{R}$ telle que $\|x\| \leq M, \forall x \in D$.

Exemple:

$D = \{x \in \mathbb{R}^2 \mid x_1 + x_2 \leq 1, \}$ est un domaine fermé et borné.

$D = \{x \in \mathbb{R}^2 \mid x_1 + x_2 < 1, \}$ est un domaine borné mais pas fermé.

Théorème 3.1

Si l'ensemble D est fermé et borné, et si la fonction f est continue sur D , alors il existe un minimum global atteint en un point de D et un maximum global atteint en un point de D .

Ou autrement, Si l'ensemble D est fermé et non vide, et si la fonction f est continue sur \mathbb{R}^n , et si l'une des conditions suivantes est vérifiée:

- D est borné
- f est coercive

Alors le problème (P2) admet au moins une solution.

Si de plus, f est strictement convexe (ex: quadratique) et D est convexe (les $g_i(x)$ convexe et les $h_i(x)$ sont affines) **alors** le problème (P2) admet une solution **unique**.

Théorème 3.2 (Condition nécessaire du premier ordre)

Si f est une fonction (Gâteaux) différentiable et si D est un convexe fermé, alors toute solution x^* de (P2) vérifie la condition nécessaire d'optimalité du premier ordre :

$$\forall x \in D, \quad \nabla f(x^*)^t(x - x^*) \geq 0$$

Théorème 3.3 (Condition nécessaire et suffisante du premier ordre)

Si f est une fonction convexe (Gâteaux) différentiable et si D est un convexe fermé, alors x^* est une solution de (P2) si et seulement si

$$\forall x \in D, \quad \nabla f(x^*)^t(x - x^*) \geq 0$$

Le problème de minimisation associé à (P2) est appelé problème de programmation **non** linéaire si au moins l'une des fonctions g, h ou f est non affine. Si toutes ces fonctions sont affines, alors on a un problème de programmation linéaire.

Remarque importante:

Une première observation pour le problème de minimisation associé à (P2) est que l'annulation du gradient à une solution n'est plus nécessaire en présence de contrainte. Par exemple pour le problème d'optimisation $x^* = \operatorname{argmin}(x^2 + x + 1)$ pour $x \in [0, 1]$ dont la seule solution est $x^* = 0$, alors que sa fonction n'a pas un gradient nul en x^* . Bien sûr cela provient du fait que le minimum est atteint au bord du domaine.

Multiplicateur de Lagrange

1) **Le cas de contrainte d'égalité:** (f : de $\Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ et h : de $\mathbb{R}^n \rightarrow \mathbb{R}^q, h_i: \mathbb{R}^n \rightarrow \mathbb{R}$)

(P1) $\operatorname{Min}f(x)$ tel que $x \in D$ Avec $D = \{x \in \mathbb{R}^n | h(x) = k\}$

Lagrange a montré qu'à l'optimalité, le vecteur gradient de la fonction objective f doit être perpendiculaire à la surface de niveau de la contrainte.

Condition nécessaire de premier ordre de Lagrange

Si x^* est un minimum local de la fonction f dans D , et si $\nabla h(x^*) \neq 0$, alors $h(x^*) = k$, et de plus il existe un scalaire $\lambda \in \mathbb{R}$ pour lequel

$$\begin{cases} \nabla f(x^*) = \lambda \nabla h(x^*) \\ h(x^*) = k \end{cases} \quad \text{si } q = 1$$

pour $q > 1$ alors $\lambda \in \mathbb{R}^q$ et la condition devient:

$$\begin{cases} \nabla f(x^*) = \sum_1^q \lambda_i \nabla h_i(x^*) \\ h_j(x^*) = k_j \end{cases}$$

Un point x^* satisfaisant cette condition est appelé un point critique.

Définition:

On appelle Lagrangien du problème de minimisation (P1) l'application \mathcal{L} : de $\Omega \times \mathbb{R}^q \rightarrow \mathbb{R}$ définie par:

$$\forall (x, \lambda) \in \Omega \times \mathbb{R}^q, \quad \mathcal{L}(x, \lambda) = f(x) + \lambda^t h(x)$$

Ou encore

$$\forall x \in \Omega, \forall \lambda_1, \dots, \lambda_q \in \mathbb{R}, \quad \mathcal{L}(x, \lambda_1, \dots, \lambda_q) = f(x) + \sum_1^q \lambda_i h_i(x)$$

Théorème 3.4

Soit x^* une solution de (P1), on suppose que les fonctions $f(x)$ et les $h_i(x)$ sont de classe C^1 et que les vecteurs $\nabla h_i(x^*)$ sont linéairement indépendants alors

$$\forall \lambda \in \mathbb{R}^q, \quad \nabla_x f(x^*, \lambda) = 0$$

c.à.d. qu'il existe des réels $\lambda_1, \dots, \lambda_q$ tels que

$$\nabla f(x^*) + \sum_{i=1}^q \lambda_i \nabla h_i(x^*) = 0$$

Cette équation est appelée équation d'Euler-Lagrange et les nombres $\lambda_1, \dots, \lambda_q$ sont appelés multiplicateurs de Lagrange.

Remarque:

La condition de Lagrange n'est qu'une condition nécessaire à l'existence d'un extremum local pour un problème sous contrainte égalitaire qui généralise l'équation d'Euler. Plus généralement l'étude des extrema de f sur D se ramène à l'étude d'extrema **sans** contrainte d'une fonction appelée le Lagrangien du problème. Elle ne permet pas de déterminer si une solution trouvée est bien un extremum local. Une amélioration du critère peut être obtenue en s'aidant de la convexité et des conditions du second ordre. Le problème que les conditions du second ordre doivent être vérifiées par rapport à l'espace tangent du domaine D et cela est difficile. En effet, l'inexploitation des conditions du second ordre provient essentiellement de la difficulté de manipuler ces conditions dans l'espace tangent du domaine D .

Théorème 3.5

Si Ω est un ouvert convexe, si f est différentiable et convexe et les $h_i(x)$ pour $i=1, \dots, q$ sont affines, alors x^* est un minimum global de f sur $D = \{x \in \mathbb{R}^n | h_i(x) = 0, p = 1, \dots, q\}$ **si et seulement si** il existe des réels $\lambda_1, \dots, \lambda_q$ tels que:

$$\nabla f(x^*) + \sum_{i=1}^q \lambda_i \nabla h_i(x^*) = 0$$

Exemple 1 : (fonction f et h de $\mathbb{R}^2 \rightarrow \mathbb{R}$)

$$\text{Min}(3x_1 - 2x_2) \text{ tel que } x_1^2 + 2x_2^2 = 44$$

La condition nécessaire d'optimalité fait en sorte que l'on doit résoudre le système:

$$h(x^*) = k \quad : \quad (x_1^*)^2 + 2(x_2^*)^2 = 44$$

$$\frac{\partial f(x^*)}{\partial x_1} = \lambda \frac{\partial h(x^*)}{\partial x_1} \quad : \quad 3 = 2\lambda x_1^*$$

$$\frac{\partial f(x^*)}{\partial x_2} = \lambda \frac{\partial h(x^*)}{\partial x_2} \quad : \quad -2 = 4\lambda x_2^*$$

Les solutions de ce système sont:

$$\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 6 \\ -2 \\ \frac{1}{4} \end{pmatrix} \text{ avec } f \begin{pmatrix} 6 \\ -2 \\ \frac{1}{4} \end{pmatrix} = 22.$$

$$\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} -6 \\ 2 \\ -\frac{1}{4} \end{pmatrix} \text{ avec } f \begin{pmatrix} -6 \\ 2 \\ -\frac{1}{4} \end{pmatrix} = -22.$$

Il n'y a pas d'autres points critiques et donc le théorème ? nous assure que $\begin{pmatrix} 6 \\ -2 \\ \frac{1}{4} \end{pmatrix}$ est le maximum global et que $\begin{pmatrix} -6 \\ 2 \\ -\frac{1}{4} \end{pmatrix}$ est le minimum global.

Interprétation de la valeur de λ

Lorsque nous résolvons un problème d'optimisation on obtient une valeur de λ en plus de la solution optimale. Cette valeur peut être interprétée de la façon suivante. Considérons la fonction d'une variable $v(k)$ qui, pour un k donné retourne la valeur optimale du problème.

$\text{Min}(x)$ tel que $h(x) = k$

Comment est-ce que la fonction v varie lorsque la valeur k varie ? Pour un k donné, définissons $x(k) \in \mathbb{R}^n$ comme étant la solution optimale du problème ci-dessus, c'est-à-dire,

$$v(k) = f(x(k)) \quad \text{et} \quad h(x(k)) = k$$

Dans ce cas $v(k)$ est une fonction qui donne la valeur minimale de f en fonction de k .

Observons aussi que $\frac{\partial f(x)}{\partial x_i} = \lambda \frac{\partial h(x)}{\partial x_i}$ pour $i=1, 2, \dots, n$. Pour mesurer la variation de $v(k)$, on calcule:

$$\begin{aligned} v'(k) &= \frac{\partial v}{\partial k} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial k} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial k} + \dots \dots \dots \frac{\partial f}{\partial x_n} \frac{\partial x_n}{\partial k} \\ &= \lambda \frac{\partial h}{\partial x_1} \frac{\partial x_1}{\partial k} + \lambda \frac{\partial h}{\partial x_2} \frac{\partial x_2}{\partial k} + \dots \dots \dots \lambda \frac{\partial h}{\partial x_n} \frac{\partial x_n}{\partial k} \\ &= \lambda \left(\frac{\partial h}{\partial x_1} \frac{\partial x_1}{\partial k} + \frac{\partial h}{\partial x_2} \frac{\partial x_2}{\partial k} + \dots \dots \dots \frac{\partial h}{\partial x_n} \frac{\partial x_n}{\partial k} \right) = \lambda \frac{\partial h}{\partial k} = \lambda \end{aligned}$$

Et donc le multiplicateur de Lagrange λ représente le taux de variation de la valeur optimale v lorsque k augmente.

On peut alors approcher la fonction v par son polynôme de Taylor de degré un $P_1(k)$ autour de k_0 :

$$v(k) \approx P_1(k) \approx v(k_0) + v'(k_0)(k - k_0) = v(k_0) + \lambda(k - k_0)$$

Suite de l'exemple 1 précédent:

Reprenons le dernier exemple. Estimez la valeur optimale du problème d'optimisation où la contrainte $x_1^2 + 2x_2^2 = 44$ est remplacée par $x_1^2 + 2x_2^2 = 45$.

On a ici $k_0 = 44$ et $k = 45$, au minimum on avait $\lambda = v'(44) = -\frac{1}{4}$ et $f = v(44) = -22$ Donc la valeur optimale serait approximativement:

$$v(45) = \begin{cases} \text{Min}(3x_1 - 2x_2) \\ \text{tel que } x_1^2 + 2x_2^2 = 45 \end{cases} \approx v(45) \approx v(44) + v'(44)(45 - 44) = -22.25$$

Exemple 2:

Soit le problème d'optimisation suivant:
 $\text{Min}(x_1^2 - x_2)$ tel que $x_1^2 + x_2^2 = 1$

Le Lagrangien s'écrit :

$$f(x_1, x_2, \lambda) = x_1^2 - x_2 + \lambda(x_1^2 + x_2^2 - 1)$$

La condition nécessaire d'ordre 1 est donnée par :

$$\begin{pmatrix} 2x_1 \\ -1 \end{pmatrix} + \lambda \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} = 0$$

Pour résoudre ce système, nous considérons les deux cas ci-dessous:

- $x_1 = 0$, alors $x_2 = \mp 1$
- $x_1 \neq 0$, alors $\lambda = -1$ et $x_2 = -1/2$ on en déduit que $x_1 = \mp \frac{\sqrt{3}}{2}$.

Il suffit maintenant de vérifier la condition d'indépendance linéaire, elle revient ici au non nullité du gradient de la fonction définissant la contrainte, ce qui est aisé de vérifier.

En conclusion, les quatre points $(\pm\sqrt{3}/2, -1/2)$, $(0, \pm 1)$ sont candidats. Il est facile de montrer que les deux premiers correspondent à un maximum absolu et que parmi les deux derniers, $(0, 1)$ correspond à un minimum absolu et $(0, -1)$ à un minimum relatif.

Remarque:

Il est crucial de vérifier la condition d'indépendance linéaire des gradients des fonctions définissant les contraintes. Cette condition est naturelle car elle écarte le cas de contraintes redondantes. Considérons en effet la minimisation de $x_1 + x_2^2$ sous la contrainte $x_1^3 - x_2^2 = 0$ dans le cas d'une unique contrainte, l'indépendance signifie la non annulation. On voit facilement que $(0, 0)$ est l'unique solution. Pourtant on n'a pas de condition d'ordre 1 :

$$\nabla_{x,\lambda} f(x_1, x_2, \lambda) = \begin{pmatrix} 1 \\ 2x_2 \end{pmatrix} + \lambda \begin{pmatrix} 3x_1^2 \\ -2x_2 \end{pmatrix} = 0$$

En $x = (0, 0)$, la condition d'ordre 1 fournit les deux égalités $1 = 0$ et $0 = 0$, ce qui est impossible.

Exemple 3:

Considérons le problème géométrique suivant : parmi tous les parallélépipèdes rectangle de surface unité, quel est celui de volume maximal ?

Notons x, y, z les longueurs des trois cotés du parallélépipède, ce sont des nombres strictement positifs. Son volume est $V(x, y, z) = xyz$ tandis que sa surface est $S(x, y, z) = 2(xy + yz + zx)$.

La question est de maximiser V sous la contrainte $S(x, y, z) = 1$. Introduisons le Lagrangien du problème :

$$f(x, y, z, \lambda) = V(x, y, z) + \lambda(S(x, y, z) - 1)$$

En un éventuel point (x, y, z) solution de ce problème d'optimisation sous contraintes, la condition d'extrémalité de Euler-Lagrange s'écrit alors

$$\exists \lambda \in \mathbb{R}, \quad \nabla V(x, y, z) + \lambda \nabla(S(x, y, z) - 1) = 0$$

c.à.d.

$$\begin{aligned} yz + 2\lambda(y + z) &= 0, \\ xz + 2\lambda(x + z) &= 0, \\ xy + 2\lambda(x + y) &= 0. \end{aligned}$$

On en déduit nécessairement que $\lambda \neq 0$ puisque $x(y + z) = y(x + z) = z(x + y)$.

En particulier, la première égalité implique que $xz = yz$ soit $x = y$ comme $z > 0$. La seconde implique que $yx = zx$ et donc que $y = z$ car $x > 0$. Finalement, on aura que $x = y = z$, le parallélépipède est alors un cube. Comme sa surface est $S(x, x, x) = 6x^2 = 1$, son coté est de longueur $1/\sqrt{6}$ puis on calculera le multiplicateur de Lagrange λ .

Note:

Notons que si f et h sont de classe \mathcal{C}^2 , les conditions nécessaires et suffisantes du second ordre établies dans le cadre de l'optimisation sans contraintes se généralisent en des conditions analogues portant ici sur la restriction de la dérivée seconde du lagrangien à l'espace tangent.

Rappelons que le lagrangien et ses dérivées sur x sont donnés par:

$$\begin{aligned} \mathcal{J}(x, \lambda) &= f(x) + \lambda^t h(x) \\ \nabla_x \mathcal{J}(x, \lambda) &= \nabla f(x) + \nabla h(x)\lambda \\ \nabla_x^2 \mathcal{J}(x, \lambda) &= \nabla^2 f(x) + \sum_1^q \lambda_i \nabla^2 h_i(x) \end{aligned}$$

Exemple 4:

Soit le problème d'optimisation suivant :

$$\min(x_1 - x_2 + x_3) \text{ tel que } \begin{cases} x_1^2 + x_2^2 + x_3^2 = 1 \\ x_1^2 + (x_2 - 1)^2 + (x_3 - 2)^2 = 4 \end{cases}$$

La condition nécessaire d'optimalité fait en sorte que l'on doit résoudre le système:

$$\begin{cases} h_1(x^*) = k_1 \\ h_2(x^*) = k_2 \\ \frac{\partial f(x^*)}{\partial x_1} = \lambda_1 \frac{\partial h_1(x^*)}{\partial x_1} + \lambda_2 \frac{\partial h_2(x^*)}{\partial x_1} \\ \frac{\partial f(x^*)}{\partial x_2} = \lambda_1 \frac{\partial h_1(x^*)}{\partial x_2} + \lambda_2 \frac{\partial h_2(x^*)}{\partial x_2} \\ \frac{\partial f(x^*)}{\partial x_3} = \lambda_1 \frac{\partial h_1(x^*)}{\partial x_3} + \lambda_2 \frac{\partial h_2(x^*)}{\partial x_3} \end{cases} \equiv \begin{cases} (x_1^*)^2 + (x_2^*)^2 + (x_3^*)^2 = 1 \\ (x_1^*)^2 + (x_2^* - 1)^2 + (x_3^* - 2)^2 = 4 \\ 1 = 2\lambda_1 x_1^* + 2\lambda_2 x_1^* \\ -1 = 2\lambda_1 x_2^* + 2\lambda_2 (x_2^* - 1) \\ 1 = 2\lambda_1 x_3^* + 2\lambda_2 (x_3^* - 2) \end{cases}$$

La troisième équation assure que $x_1^* \neq 0$, et donc on peut diviser par cette variable pour obtenir $2\lambda_1 + 2\lambda_2 = \frac{1}{x_1^*}$. En substituant dans les quatrième et cinquième équations, on obtient:

$$\begin{cases} -1 = \frac{x_2^*}{x_1^*} - 2\lambda_2 \\ 1 = \frac{x_3^*}{x_1^*} - 4\lambda_2 \end{cases} \Rightarrow 4\lambda_2 = \frac{x_3^*}{x_1^*} - 1 = \frac{2x_2^*}{x_1^*} + 2$$

À partir des deux premières équations on obtient $x_2^* + 2x_3^* = 1$, en combinant (en fixant $x_3^* = \frac{1}{2} - \frac{x_2^*}{2}$) avec le résultat précédent on obtient: $x_2^* = \frac{1}{5} - \frac{6x_1^*}{5}$. Les troisième et quatrième équations donnent les multiplicateurs:

$$\lambda_1 = \frac{1}{10} + \frac{2}{5x_1^*} \quad \text{et} \quad \lambda_2 = -\frac{1}{10} + \frac{1}{10x_1^*}$$

La première équation se simplifie en $280(x_1^*)^2 = 80$ et admet les deux solutions $\pm\sqrt{2/7}$ et on obtient les valeurs des autres variables. En effet, pour $x_1^* = \sqrt{2/7}$ on aura x_2^* et x_3^* puis on calcule $f(x^*) \approx 1.8967$. Alors que pour $x_1^* = -\sqrt{2/7}$, on obtient $f(x^*) \approx -1.0967$.

Puisqu'il n'y a pas d'autres points critiques donc la première solution représente le maximum global et la deuxième le minimum global.

2) **Le cas de contrainte d'inégalité:** (f : de $\Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ et g : de $\mathbb{R}^n \rightarrow \mathbb{R}^p$ et g_i : de $\mathbb{R}^n \rightarrow \mathbb{R}$)

(P2) $\text{Min}f(x)$ tel que $x \in D$ Avec $D = \{x \in \mathbb{R}^n | g(x) \leq k\}$

Dans ce cas de contrainte d'inégalité large et pour $p=1$, nous distinguons deux situations:

- Le minimum local x^* est dans D ($g(x) < k$), nous avons alors $\nabla f(x^*) = 0$.
- Le minimum local x^* est sur la frontière de D (c'est le cas précédent de contrainte égalitaire où $g(x)=k$), alors le gradient $\nabla f(x^*)$ satisfait

$$\begin{cases} \nabla f(x^*) = \lambda \nabla g(x^*) \\ g(x^*) = k \end{cases}$$

pour $p > 1$ alors $\lambda \in \mathbb{R}^p$ et la condition devient:

$$\begin{cases} \nabla f(x^*) = \sum_{i=1}^p \lambda_i \nabla g_i(x^*) \\ g_j(x^*) = k_j \end{cases}$$

De la même manière que le cas de contrainte égalitaire, on définit le Lagrangien du problème (P2) Comme suit:

Définition:

On appelle Lagrangien du problème de minimisation (P2) l'application \mathcal{L} : de $\Omega \times \mathbb{R}^p \rightarrow \mathbb{R}$ définie par:

$$\forall (x, \mu) \in \Omega \times \mathbb{R}^p, \quad \mathcal{L}(x, \mu) = f(x) + \mu^t g(x)$$

Ou encore

$$\forall x \in \Omega, \forall \mu_1, \dots, \mu_p \in \mathbb{R}, \quad \mathcal{L}(x, \mu_1, \dots, \mu_p) = f(x) + \sum_{i=1}^p \mu_i g_i(x)$$

Le résultat suivant fournit la condition nécessaire d'ordre 1 dans le cas des contraintes de type inégalités :

Théorème 3.6 (Conditions de Kuhn et Tucker)

Soit x^* une solution du problème (P2). On suppose que les fonctions f, g_i sont de classe \mathcal{C}^1 et que la condition suivante est vérifiée:

i. $\exists d \in \mathbb{R}^n, g_i(x^*) = 0 \Rightarrow d^t \nabla g_i(x^*) < 0$

Alors il existe des réels μ_1, \dots, μ_p tel que:

ii. $\mu_i \geq 0$

iii. $\mu_i g_i(x^*) = 0$

iv. $\nabla f(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$

- La condition (i) est plus faible qu'une condition d'indépendance linéaire. On appelle (i) hypothèse de qualification des contraintes.
- Les coefficients μ_1, \dots, μ_p sont appelés multiplicateurs de Kuhn et Tucker.
- La condition (iii) est une relation d'exclusion. Elle signifie que si la contrainte n'est pas serrée (i.e. $g_i(x^*) < 0$), alors $\mu_i = 0$: elle n'intervient pas dans la condition d'ordre 1.
- La condition (iv) s'écrit en termes de Lagrangien :

$$\nabla_x \mathcal{L}(x^*, M) = 0, \text{ avec } M = \mu_1, \dots, \mu_p$$

Remarque:

Si on a affaire à un problème de maximisation, on considère $-f$ pour se ramener à un problème de minimisation, plutôt que de changer le signe des multiplicateurs.

Exemple 1:

Soit à maximiser $f(x) = (x_1 - 1)^2 + (x_2 - 2)^2$ tel que $x_1^2 + x_2^2 \leq 45$ c.à.d.

$$(g(x) = x_1^2 + x_2^2 - 45 \leq 0).$$

Notons que le domaine défini par $g(x)$ est fermé et borné alors il contient à la fois le minimum et le maximum globaux. Nous commençons d'abord par identifier tous les points où le gradient de f s'annule, et nous retiendrons seulement ceux appartenant au domaine. Puis nous allons appliquer la méthode de Lagrange où nous remplacerons l'inégalité par une égalité. Enfin, le minimum et le maximum global seront alors parmi les points énumérés. Il suffira donc d'évaluer f en tous ces points.

- Étape 1: Points critiques à l'intérieur strict du D définie par $g(x)$

Nous obtenons les points critiques par: $\nabla f(x) = 0$

$$\frac{\partial f}{\partial x_1} = 2(x_1 - 1) = 0 \rightarrow x_1 = 1$$

$$\frac{\partial f}{\partial x_2} = 2(x_2 - 2) = 0 \rightarrow x_2 = 2$$

En ce point $g(1,2)=5 < 45$ et le point critique $(1, 2)$ se trouve à l'intérieur strict du domaine D .

- Étape 2: Points critiques sur la frontière de D

Pour ces points, on utilisera les multiplicateurs de Lagrange:

$$\begin{cases} g(x) = k \\ \frac{\partial f(x)}{\partial x_1} = \lambda \frac{\partial g(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} = \lambda \frac{\partial g(x)}{\partial x_2} \end{cases} \equiv \begin{cases} x_1^2 + x_2^2 = 45 \\ 2(x_1 - 1) = 2\lambda x_1 \\ \frac{\partial f(x)}{\partial x_2} = 2\lambda x_2 \end{cases}$$

Les solutions sont $(x_1, x_2) = (3, 6)$ et $(x_1, x_2) = (-3, -6)$, qui sont alors deux points critiques.

- Étape 3 : Évaluation de f aux points critiques.

$$f(1, 2) = 0; f(3, 6) = 20; f(-3, -6) = 80$$

Ainsi la valeur minimale de f est 0 et la valeur maximale est de 80.

Exemple 2:

Soit le problème de minimiser x_1 sur l'ensemble $D = \{x \in \mathbb{R}^2 | x_2 \geq 0 \text{ et } x_2 \leq (x_1 + 1)^3\}$

Le Lagrangien du problème est :

$$J(x, \mu_1, \mu_2) = x_1 - \mu_1 x_2 + \mu_2 (x_2 - (x_1 + 1)^3)$$

$$\nabla_x J(x, \mu_1, \mu_2) = \begin{pmatrix} 1 - 3\mu_2(x_1 + 1)^2 \\ -\mu_1 + \mu_2 \end{pmatrix}$$

La condition d'ordre 1 s'écrit :

$$1 = 3\mu_2(x_1 + 1)^2 \text{ et } \mu_1 = \mu_2$$

Alors que les conditions d'exclusion fournissent :

$$\mu_1 x_2 = 0 \quad \text{et} \quad \mu_2 (x_2 - (x_1 + 1)^3) = 0$$

- Si $\mu_1 = 0 \Rightarrow \mu_2 = 0 \Rightarrow 1 = 0$ ce qui est impossible
- Sinon, si $x_2 \Rightarrow x_1 = -1$ puisque $\mu_2 \neq 0$. Donc le seul point critique est $(-1, 0)$.
- Il reste à vérifier l'hypothèse de qualification des contraintes :

En notant $h_1(x) = -x_2$ et $h_2(x) = x_2 - (x_1 + 1)^3$, on aura:

$$\nabla h_1(-1, 0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \quad \text{et} \quad \nabla h_2(-1, 0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Et justement, elles ne sont pas vérifiées ! Supposons qu'elles le soient : il existerait un vecteur $d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ tel que

$$d^t \nabla h_1(-1, 0) = -d_2 < 0 \quad \text{et} \quad d^t \nabla h_2(-1, 0) = d_2 < 0 \quad \text{ce qui impossible}$$

Malgré que le point $(-1, 0)$ est bien le point recherché, le théorème de Kuhn et Tucker n'est pas applicable.

Exemple 3: (cas quadratique)

3) Conditions d'optimalité en présence de contraintes d'égalité et d'inégalité

On considère maintenant le problème général où les contraintes peuvent être de type égalitaire et/ou inégalitaire, là les conditions de Lagrange se généralisent aux conditions de Karush-Kuhn-Tucker.

Soit Ω un ouvert non vide de \mathbb{R}^n ; soit $f: \Omega \rightarrow \mathbb{R}$ une application différentiable et soient

$h_i(x)$ avec $i = 1, 2, \dots, p$; $g_i(x)$ avec $i = 1, 2, \dots, q$ des applications de classe \mathcal{C}^1 ($p, q \geq 0$).

L'objectif est de minimiser $f(x)$ sur l'ensemble:

$$D = \{x \in \mathbb{R}^n \mid h_i(x) = 0, \quad i = 1, 2, \dots, p \quad \text{et} \quad g_i(x) \leq 0, \quad i = 1, 2, \dots, q\}$$

Ainsi, nous nous trouvons en face du problème d'optimisation suivant:

$$(P3) \quad \text{Minimiser } f(x) \text{ tel que } \begin{cases} h_i(x) = 0, \quad i = 1, 2, \dots, p \\ g_i(x) \leq 0, \quad i = 1, 2, \dots, q \\ x \in \mathbb{R}^n \end{cases} \equiv \text{Minimiser } f(x) \text{ tel que } x \in D$$

On définit le Lagrangien pour le problème (P3) de la même manière que pour un problème 'optimisation sous contraintes égalitaires.

Définition:

On appelle Lagrangien du problème d'optimisation sous contraintes égalitaires et inégalitaires (P3) l'application : $\mathcal{L}: \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}$

$$(x, \lambda, \mu) \rightarrow \mathcal{L}(x, \lambda, \mu) = f(x) + \sum_1^p \lambda_i h_i(x) + \sum_1^q \mu_i g_i(x)$$

Lorsque toutes les applications sont différentiables, le vecteur gradient du Lagrangien en $x \in \mathbb{R}^n$ est donné par:

$$\nabla_x \mathcal{L}(x, \lambda, \mu) = \nabla f(x) + \sum_1^p \lambda_i \nabla h_i(x) + \sum_1^q \mu_i \nabla g_i(x)$$

La condition nécessaire du premier ordre est donnée par:

$$\exists \lambda \in \mathbb{R}^p, \mu \in \mathbb{R}^q \text{ tel que } \nabla_x \mathcal{L}(x, \lambda, \mu) = 0$$

On définit aussi, lorsque toutes les applications sont deux fois différentiables, la matrice Hessienne du Lagrangien en $x \in \mathbb{R}^n$ par:

$$\nabla_{x,d}^2 \mathcal{L}(x, \lambda, \mu) = \nabla^2 f(x) + \sum_1^p \lambda_i \nabla^2 h_i(x) + \sum_1^q \mu_i \nabla^2 g_i(x)$$

L'extension du théorème précédent au cas général est directe.

Théorème 3.7 (Conditions nécessaires de Karush-Kuhn-Tucker)

Soit x^* une solution du problème (P3). On suppose que les fonctions f , h_i et g_i sont de classe \mathcal{C}^1 et que les conditions de qualification des contraintes suivantes sont vérifiées:

- $\exists d \in \mathbb{R}^n$, $d^t \nabla h_i(x^*) = 0$ et $g_i(x^*) = 0 \Rightarrow d^t \nabla g_i(x^*) < 0$
- Et les vecteurs $\nabla h_i(x^*)$ avec $i = 1, \dots, p$ sont linéairement indépendants

Alors il existe des réels $\lambda_1, \dots, \lambda_p$ et des réels μ_1, \dots, μ_p tels que:

- i. $\mu_i \geq 0$
- ii. $\mu_i g_i(x^*) = 0$
- iii. $h_i(x^*) = 0$
- iv. $\nabla f(x^*) + \sum_1^p \lambda_i \nabla h_i(x^*) + \sum_1^q \mu_i \nabla g_i(x^*) = 0$

En effet, si x^* est un point régulier pour les contraintes h et g , les conditions de qualification des contraintes ainsi que (ii & iii) sont vérifiées.

Les λ_i, μ_i sont appelés les multiplicateurs de Lagrange-KKT ou multiplicateurs de Lagrange généralisés.

Remarque:

Dans le cas d'une contrainte insaturée $g_i(x^*) \neq 0$, le coefficient de Lagrange-KKT correspondant est nul : $\mu_i = 0$, c'est-à-dire que cette contrainte ne compte pas. Lorsque toutes les contraintes inégalitaires sont insaturées en x^* , le domaine D ainsi formé est défini uniquement par les contraintes égalitaires et on retrouve les conditions de Lagrange.

Théorème 3.8 (CNS pour le cas convexe)

Si les fonctions f , h et g sont de classe \mathcal{C}^1 et que f et g sont convexes, h est affine; **alors** les conditions du théorème précédent deviennent nécessaires et suffisantes pour que x^* soit une solution globale du problème (P3).

Exemple:

Soit le problème de minimiser x_1^2 sur l'ensemble $D = \{x \in \mathbb{R}^2 | x_2 = x_1 \text{ et } x_1^2 + x_2^2 \leq 2\}$

Le Lagrangien du problème est :

$$\mathcal{L}(x, \lambda, \mu) = x_1^2 + \lambda(x_2 - x_1) + \mu(x_1^2 + x_2^2 - 2)$$

$$\nabla_{x,d} \mathcal{L}(x, \lambda, \mu) = \begin{pmatrix} 2x_1^* - \lambda + 2\mu x_1^* \\ \lambda + 2\mu x_2^* \end{pmatrix}$$

La condition d'ordre 1 s'écrit :

$$\lambda = (2 + 2\mu)x_1^* \text{ et } \lambda = -2\mu x_2^*$$

Alors que les conditions d'exclusion fournissent :

$$x_2^* = x_1^* \text{ et } \mu((x_1^*)^2 + (x_2^*)^2 - 2) = 0, \mu \geq 0$$

On en déduit que $(1 + 2\mu)x_1^* = 0 \Rightarrow x_1^* = 0$ puisque $\mu \geq 0$, $x_2^* = x_1^* \Rightarrow x_2^* = 0$.
La condition de qualification des contraintes est donc vérifiée.

Remarque : La méthode des multiplicateurs de Lagrange ne peut pas être utilisée dans tous les cas. Notamment lorsque le problème d'optimisation possède des contraintes de non-négativité ou lorsque la fonction n'est pas dérivable, cette méthode n'est pas adaptée.

Méthodes de Programmation Quadratique Successive (SQP)

Rappelons-nous que dans un problème d'optimisation en présence de contraintes, le Lagrangien joue le rôle de la fonction objectif. Ce que nous avons obtenu jusqu'à maintenant comme solution aux problèmes d'optimisation avec contraintes égalitaires et/ou inégalitaires, est une solution x^* définie seulement comme étant un point critique du Lagrangien et rien ne prouve que le Lagrangien présente un minimum en ce point. Une méthode de descente particulière a été développée pour exploiter le résultat précédent et faire la suite du travail. L'idée essentielle consiste à résoudre une succession de problèmes quadratiques avec contraintes linéaires (ces problèmes sont relativement simples à résoudre) qui sont des approximations du problème de départ.

Commençons-nous d'abord par le cas de problème à contraintes égalitaires suivant:

$$(P1) \quad \text{Minf}(x) \text{ tel que } x \in D \text{ Avec } D = \{x \in \mathbb{R}^n \mid h_i(x) = 0 ; i = 1, \dots, p\}$$

Étant donnée un itéré x_k on cherche $x_{k+1} = x_k + \rho_k d_k$.

Où $d_k \in \mathbb{R}^n$ est une direction de descente et $\rho_k > 0$ est le pas.

Commençons par faire une approximation des contraintes h grâce en utilisant la formule de Taylor au premier ordre (pour $\rho_k = 1$):

$$h_i(x_k + d_k) \cong h_i(x_k) + \nabla h_i(x_k)^t d_k + O^2(d_k)$$

Si on néglige les termes d'ordre supérieur ou égal à 2, on définit la direction d_k comme étant la direction permettant d'assurer que $h_i(x_k + d_k) \cong 0$. Plus précisément, on pose

$$h_i(x_k) + \nabla h_i(x_k)^t d_k \cong 0$$

C.à.d:

$$Dh(x_k)d_k = -h(x_k)$$

Où $Dh(x_k)$ est la Jacobienne de h en x_k . Cette relation correspond à une linéarisation des contraintes au voisinage de x_k : c'est un système linéaire.

Par ailleurs, il faudrait que x_{k+1} diminue la valeur du Lagrangien. De manière similaire on va faire une approximation du Lagrangien $\mathcal{L}(x, y) = f(x) + y^t h(x)$: elle sera quadratique cette fois, puisque le point cherché est un point critique et qu'on ne peut se contenter d'une approximation du premier ordre.

$$\mathcal{L}(x_k + d_k, \lambda) = \mathcal{L}(x_k, \lambda) + \nabla \mathcal{L}(x_k, \lambda)^t d_k + \frac{1}{2} d_k^t D_{xx}^2 \mathcal{L}(x_k, \lambda) d_k + O^3(\|d_k\|^3)$$

Si on néglige les termes d'ordre supérieur ou égal à 3, on voit qu'il faut minimiser

$$\nabla \mathcal{L}(x_k, \lambda)^t d_k + \frac{1}{2} d_k^t D_{xx}^2 \mathcal{L}(x_k, \lambda) d_k$$

pour espérer minimiser le Lagrangien. On cherche donc, en fin de compte x_k solution du problème:

$$(QP)_e \quad \begin{cases} \min_{d_k} \nabla \mathcal{L}(x_k, \lambda)^t d_k + \frac{1}{2} d_k^t D_{xx}^2 \mathcal{L}(x_k, \lambda) d_k \\ Dh(x_k)d_k + h(x_k) = 0 \end{cases}$$

En effet, nous avons :

$$\begin{aligned} \nabla_{\lambda} f(x_k, \lambda)^t d_k &= \nabla f(x_k)^t d_k + \lambda^t Dh(x_k) d_k \\ &= \nabla f(x_k)^t d_k + \underbrace{\lambda^t h(x_k)}_{\text{constante}} \end{aligned}$$

Le dernier terme étant constant, il reste ensuite à déterminer le pas ρ_k et le multiplicateur λ_k à chaque itération. Il y a bien sûr, beaucoup de possibilités qui génèrent autant de variantes de la méthode.

Nous présentons ici, la méthode qui est basée sur le choix : $\rho_k = 1$.

Algorithme de la méthode SQP pour des contraintes en égalité:

1. Initialisation

K=1: choix de $x_0 \in \mathbb{R}^n$ et de $\lambda_0 \in \mathbb{R}^p$

2. Itération k

Résoudre le sous problème quadratique

$$(QP)_e \quad \begin{cases} \min_{d_k} \nabla f(x_k)^t d_k + \frac{1}{2} d_k^t D_{xx}^2 f(x_k, \lambda_k) d_k \\ Dh(x_k) d_k + h(x_k) = 0 \end{cases}$$

3. On pose $\lambda_{k+1} \in \mathbb{R}^p$ alors comme multiplicateur associé à la contrainte (en égalité) de $(QP)_e$ et $x_{k+1} = x_k + d_k$.

4. Critère d'arrêt

si $\|x^{k+1} - x^k\| < \varepsilon$ stop

Sinon $k=k+1$ et revenir à 2.

Cas de contraintes générales

Dans le cas de contraintes en égalité et inégalité le principe est le même : seule la fonction Lagrangienne change. Ainsi pour le problème

$$(P3) \quad \text{Minimiser } f(x) \text{ tel que } \begin{cases} h_i(x) = 0, & i = 1, 2, \dots, p \\ g_j(x) \leq 0, & i = 1, 2, \dots, q \\ x \in \mathbb{R}^n \end{cases} \equiv \text{Minimiser } f(x) \text{ tel que } x \in D$$

$$D = \{x \in \mathbb{R}^n \mid h_i(x) = 0, \quad i = 1, 2, \dots, p \text{ et } g_j(x) \leq 0, \quad i = 1, 2, \dots, q\}$$

Le Lagrangien vaut:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^t h(x) + \mu^t g(x), \quad \text{où } \lambda \in \mathbb{R}^p \text{ et } \mu \in \mathbb{R}^q.$$

La méthode SQP s'écrit de la même façon : on linéarise les contraintes et on fait une approximation quadratique du Lagrangien. Cela conduit à l'algorithme suivant:

Algorithme de la méthode SQP pour des contraintes générales

1. Initialisation

K=1: choix de $x_0 \in \mathbb{R}^n$ et de $\lambda_0 \in \mathbb{R}^p, \mu_0 \in \mathbb{R}^{q+}$

2. Itération k

Résoudre le sous problème quadratique

$$(QP)_e \begin{cases} \min_{\mathbb{R}^n} \left\{ \nabla f(x_k)^t d_k + \frac{1}{2} d_k^t D_{xx}^2 f(x_k, \lambda_k, \mu_k) d_k \right. \\ \quad Dh(x_k) d_k + h(x_k) = 0 \\ \quad Dg(x_k) d_k + h(x_k) \leq 0 \end{cases}$$

3. On pose $\lambda_{k+1} \in \mathbb{R}^p$ alors comme multiplicateur associé à la contrainte (en égalité) de $(QP)_e$ et $\mu_{k+1} \in \mathbb{R}^q$ comme multiplicateur associé à la contrainte (en inégalité). $x_{k+1} = x_k + d_k$.

4. Critère d'arrêt

si $\|x^{k+1} - x^k\| < \varepsilon$ stop

Sinon $k=k+1$ et revenir à 2.

Il existe pour cet algorithme même un résultat de convergence.

Introduction à la théorie de la dualité:

La théorie de la dualité et en particulier celle de la dualité convexe est outil très puissant où l'idée générale est au lieu de considérer la fonction objective f , on considère le Lagrangien du problème d'optimisation \mathcal{L} puisque ce dernier englobe et la fonction objective et toutes les contraintes donc plus représentative du problème. De plus, nous avons qu'une condition nécessaire du premier ordre pour qu'un point x^* soit un minimum de f et que le x^* associé aux multiplicateurs de Lagrange soit un point critique de \mathcal{L} .

Rappelons d'abord qu'une surface décrite par une équation $f(x, y) = z$ peut être considéré comme une surface de niveau en posant $F = f(x, y) - z$

Si f est différentiable, on aura: $\nabla F = i\nabla_x f + j\nabla_y f - kz$, nous avons alors par simple dérivation l'équation du plan tangent en un point $x^0 = (x_0, y_0, z_0)$.

$$(z - z_0) = A(x - x_0) + B(y - y_0)$$

Avec

$$A = \nabla_x f(x_0, y_0) \text{ et } B = \nabla_y f(x_0, y_0)$$

Lorsque $\nabla f(x^0) = 0$ (i.e. $A = B = 0$), le point x^0 est un point stationnaire ou critique de f . Ainsi dans notre cas, on aura en un point critique $\nabla F(x^0) = -k$ ce qui signifie géométriquement que le plan tangent est horizontal en ce point. La classifications des points stationnaires revient à imaginer un paysage montagneux et consiste justement à séparer les sommets (maximums) des montagnes, les vallées (minimums) des montagnes puis les passages (points-sell) montagneux.

Définition:

Soit $f(x)$ une fonction de classe \mathcal{C}^1 et x^0 un point stationnaire de f , x^0 est appelé point-sell de f si pour toute boule $B(x^0)$ il existe des points x tels que $f(x) < f(x^0)$ et d'autres pour lesquels $f(x) > f(x^0)$.

Pour le cas monovarié, nous retrouvons la définition du point d'inflexion.

Pour notre problème d'optimisation général (P3) dont le Lagrangien est $\mathcal{L}: \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}$

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_1^p \lambda_i h_i(x) + \sum_1^q \mu_i g_i(x)$$

On aura plutôt la définition suivante:

Définition:

On appelle point-sell du Lagrangien sur $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+q}$ tout triplet $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+q}$ vérifiant l'équation:

$$\forall (x, \lambda, \mu) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+q}, \quad f(x, \lambda, \mu) \leq f(x^*, \lambda^*, \mu^*) \leq f(x, \lambda^*, \mu^*)$$

C.à.d.

- x^* est un minimum de $x \rightarrow f(x, \lambda^*, \mu^*)$ et
- (λ^*, μ^*) est un maximum de $(\lambda, \mu) \rightarrow f(x^*, \lambda, \mu)$

Autrement dit, une caractérisation du point-sell (x^*, λ^*, μ^*) est donnée par:

$$\sup_{(\lambda, \mu)} \inf_x f(x, \lambda, \mu) = f(x^*, \lambda^*, \mu^*) = \inf_x \sup_{(\lambda, \mu)} f(x, \lambda, \mu)$$

Le point-sell (x^*, λ^*, μ^*) du Lagrangien $f(x, \lambda, \mu)$ fournit une solution à notre problème d'optimisation (P3).

Théorème 3.9

Supposons que f, g et h sont de classe \mathcal{C}^1 et que le triplet $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+q}$ est un point-sell de f sur $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+q}$, alors ce triplet vérifie les conditions de Karush-

Kuhn-Tucker du théorème précédent et x^* est une solution du problème d'optimisation (P3).

Théorème 3.10 (cas convexe):

Supposons que f, g sont convexes et de classe \mathcal{C}^1 , h affine et de classe \mathcal{C}^1 ; alors ce triplet $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+q}$ est un point-sell de f sur $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+q}$ si et seulement si ce triplet vérifie les conditions de Karush-Kuhn-Tucker du théorème précédent (si et seulement si x^* est une solution du problème d'optimisation (P3)).

Exemple d'utilisation du point-sell Lagrangien en optimisation (méthode d'UZAWA):

D'après la caractérisation précédente du point-sell (solution à notre problème), nous devons aller chercher le triplet $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+q}$ vérifiant les conditions de Karush-

Kuhn-Tucker de la façon suivante:

1. Pour (λ^*, μ^*) de $\mathbb{R}^p \times \mathbb{R}^{+q}$ fixés, nous allons chercher le minimum **sans** contraintes (sur tout l'espace \mathbb{R}^n) de la fonction $x \rightarrow f(x, \lambda^*, \mu^*)$.
2. Pour x^* de \mathbb{R}^n fixé, on cherche le maximum sur $\mathbb{R}^p \times \mathbb{R}^{+q}$ (c.à.d. avec simplement des contraintes de bornes) de la fonction $(\lambda, \mu) \rightarrow f(x^*, \lambda, \mu)$

Bien sûr, on ne peut pas faire ces deux calculs simultanément; on va plutôt résoudre successivement ces deux étapes et on obtient ainsi l'algorithme d' Uzawa suivant:

Algorithme d'Uzawa

1. Initialisation

$K=0$: choix de λ^0 et μ^0

2. Itération k

$\lambda^k = (\lambda_1^k, \dots, \lambda_p^k) \in \mathbb{R}^p$ et $\mu^k = (\mu_1^k, \dots, \mu_q^k) \in \mathbb{R}^{+q}$ sont connus, puis

- i. Calcul de $x^k \in \mathbb{R}^n$ solution du problème

$$(P^k) \quad \min_x f(x, \lambda^k, \mu^k), x \in \mathbb{R}^n$$

ii. Calculs de λ^{k+1} et μ^{k+1} avec

$$\lambda_i^{k+1} = \lambda_i^k + \rho h_i(x^k), \quad i = 1, \dots, p$$

$$\mu_j^{k+1} = \max(0, \mu_j^k + \rho g_j(x^k)), \quad j = 1, \dots, q$$

où $\rho > 0$ est un réel fixé (à choisir par l'utilisateur).

3. Critère d'arrêt

si $\|x^{k+1} - x^k\| < \varepsilon$ stop

Sinon $k=k+1$ et revenir à 2.

Évidemment l'étape 2(i) revient à résoudre $\nabla f(x) + \sum_1^p \lambda_i \nabla h_i(x) + \sum_1^q \mu_j \nabla g_j(x^*) = 0$, alors que 2(ii) est très facile à effectuer. la convergence de cet algorithme est assurée pour une fonction objective de classe C^1 elliptique et h, g convexes (h au moins affine) de classe C^1 et Lipschitziennes.

Méthodes de pénalisation (extérieure):

Les méthodes de pénalisation sont très simples et souvent utilisées en pratique, elles partent du principe suivant : on remplace le problème avec contraintes.

$$(P) \quad \begin{cases} \min f(x) \\ x \in D \subseteq \mathbb{R}^n \end{cases}$$

Par un problème sans contraintes

$$(P_\varepsilon) \quad \begin{cases} \min f(x) + \frac{1}{\varepsilon} \alpha(x) \\ x \in \mathbb{R}^n \end{cases}$$

Où $\alpha: \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction de pénalisation des contraintes et $\varepsilon > 0$. Le but est de trouver des fonctions α telles que les problèmes (P) et (P_ε) soient équivalents, c'est-à-dire, tels qu'ils aient les mêmes solutions.

Dans ce cas, on dit que la pénalisation est exacte. On peut, par exemple, choisir :

$$\alpha(x) = \begin{cases} 0 & \text{si } x \in D \\ +\infty & \text{si } x \notin D \end{cases}$$

Cette fonction n'a pas de bonnes propriétés mathématiques (notamment la dérivabilité) pour qu'on puisse appliquer les techniques de résolution sans contraintes. Le principe intuitif des méthodes de pénalisation est que l'on espère que, lorsque ε devient petit, on aura tendance à satisfaire la contrainte α pour compenser.

En général, on effectue ce que l'on appelle une pénalisation dite inexacte, telle que le problème (P) à des solutions qui ne sont pas solutions de (P_ε) ; l'ensemble des solutions de (P_ε) ne couvre pas tout l'ensemble des solutions de (P).

Néanmoins, on peut trouver dans ce cas des fonctions α qui sont dérivables, ce qui permet d'utiliser les résultats de minimisation sans contraintes.

Donnons quelques exemples de fonctions de pénalisation α où la pénalisation est dite extérieure car la suite (x_ε) avec $\varepsilon > 0$ converge vers x^* en venant de l'extérieur de D.

Ici, nous supposons que α vérifie les propriétés suivantes :

- i. α est continu sur \mathbb{R}^n
- ii. $\forall x \in \mathbb{R}^n, \alpha(x) \geq 0$
- iii. $\alpha(x) = 0 \Leftrightarrow x \in D$

Nous donnons quelques exemples de fonction de pénalisation pour différentes contraintes :

- ✓ Contrainte $x \leq 0$: la fonction α est $\alpha(x) = \|x^+\|^2$, $x^+ = (x_1^+, \dots, x_n^+)$.
- ✓ Contrainte $h(x) = 0$: la fonction α est $\alpha(x) = \|h(x)\|^2$.
- ✓ Contrainte $g(x) \leq 0$: la fonction α est $\alpha(x) = \|g(x)^+\|^2$.

Théorème 3.11

Soit une fonction f continue et coercive et soit D un domaine fermé non vide. Si α vérifie (i-iii) on a alors:

- ✓ $\forall \varepsilon > 0$, (P_ε) possède au moins une solution x_ε
- ✓ la suite (x_ε) avec $\varepsilon > 0$ est bornée.
- ✓ Toute sous-suite convergente extraite de $(x_\varepsilon)_{\varepsilon > 0}$ converge vers une solution de (P) lorsque $\varepsilon \rightarrow 0$.

On obtient alors l'algorithme de pénalisation extérieure suivant:

Initialisation

$K=1$: choix de x^0 et $\varepsilon^{(1)} > 0$

Itération k

Tant que le critère d'arrêt n'est pas satisfait :

- i. Résoudre le sous-problème

$$P_{\varepsilon^{(k)}} \quad \begin{cases} \min_{x \in \mathbb{R}^n} f(x) + \frac{1}{\varepsilon^{(k)}} \alpha(x), \\ x \in \mathbb{R}^n \end{cases}$$

- ii. $k \leftarrow k + 1$, prendre $\varepsilon^{(k+1)} < \varepsilon^{(k)}$

Méthode de point intérieur:

Contraire aux méthodes de pénalisation externe qui consistent à remplacer les contraintes d'inégalité par une fonction nulle dans le domaine des contraintes et strictement positive à l'extérieur. Les méthodes de point intérieur visent à remplacer les contraintes du critère par un terme additif qui tend vers l'infini à la frontière du domaine des contraintes lorsqu'on augmente un paramètre de réglage d'adéquation aux contraintes que l'on notera ici t . Ainsi, au problème:

$$(P) \quad \begin{cases} \min_x f(x) \\ g_i(x) \leq 0 \end{cases} \quad i=1, \dots, q$$

On pourra associer le critère non contraint suivant

$$f(x) + \frac{1}{t} \sum_{i=1}^q \Psi(-g_i(x))$$

où $\Psi(z)$ est une fonction décroissante sur \mathbb{R}^+ qui présente une divergence en 0. En pratique, on cherchera à minimiser itérativement cette fonction tout en faisant croître la valeur de t . Typiquement, on prendra $\Psi(z) = \log z$. Nous reviendrons en détail sur cette fameuse méthode au dernier chapitre.

Note:

La méthode de point intérieur constitue une technique de pénalisation interne qui conduit à des algorithmes itératifs qui doivent être initialisés à l'intérieur du domaine des contraintes.

Chapitre 4

Programmation linéaire

Introduction:

La programmation linéaire est définie comme étant un cas particulier de la programmation mathématique pour laquelle la fonction objective et les contraintes sont linéaires et où généralement les variables sont supposées non-négatives. Si de plus, on impose que ces variables ne peuvent prendre que des valeurs entières, on parle de programmation linéaire entière. Il s'agit d'une classe d'optimisation dont le champ d'application est énorme notamment en économie. Ce chapitre présente une introduction à ce domaine.

En ce qui concerne les techniques d'optimisation proprement dites, elles sont nombreuses. Les premiers travaux datent depuis les années cinquante avec le développement de la programmation mathématique linéaire et non-linéaire. Notons dès à présent que la démarche intuitive qui consiste à ne faire varier qu'un seul facteur à la fois, en maintenant tous les autres constants, puis à recommencer la procédure pour chaque facteur en fixant au fur et à mesure chaque facteur à son meilleur niveau n'est pas une méthode recommandable. Son majeur inconvénient est de réclamer un grand nombre d'essais. Il existe des méthodes plus exactes, généralement plus rapides. Le principe commun à ces méthodes est de faire varier simultanément tous les facteurs intervenant dans l'optimisation ; c'est le cas de la méthode du simplexe. Il s'agit d'une méthode itérative : les résultats sont exploités au fur et à mesure de leur obtention ; ils servent à déterminer de nouvelles conditions expérimentales qui permettent de progresser dans la recherche de l'optimum.

La présentation algébrique de la méthode du simplexe permet de donner une interprétation pratique aux différentes opérations à effectuer dans la résolution du problème de programmation linéaire. Toutefois, la méthode algébrique devient rapidement laborieuse à mesure que le nombre de contraintes et le nombre de variables augmentent.

Nous allons donc s'intéresser directement à la version systématique de toutes ces opérations algébriques sous forme d'un simple tableau dit « tableau du simplexe ». Ceci permettra de résoudre plus facilement les problèmes ayant beaucoup plus de variables et de contraintes. Nous ne donnerons pas ici les preuves des résultats à énoncer, nous en tenons aux idées directrices.

L'approche utilisée pour résoudre ce type de problèmes sera divisée en deux étapes principales :

- a) **La modélisation** du problème sous forme d'équations ou d'inéquations linéaires qui permettra ainsi de bien identifier et structurer les contraintes que doivent respecter les variables du modèle . Évidemment, on doit définir l'apport de chaque variable à la fonction à optimiser.
- b) La détermination de l'**optimum mathématique** à l'aide de certaines techniques propres à la programmation linéaire.

Ensemble-Solution d'inéquation ou système d'inéquations linéaires:

Définition: Une **inéquation linéaire** est une expression de la forme :

$$a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$$

où x_i sont les variables (ou inconnues), les a_i sont les coefficients des variables, b est une constante et n est le nombre d'inconnues.

Définition: On appelle solution d'une **inéquation linéaire** de la forme :

$$a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$$

Tout n-tuplet $(s_1; s_2; \dots; s_n)$ qui vérifie:

$$a_1s_1 + a_2s_2 + \dots + a_ns_n \leq b$$

Cas n=2:

Ainsi le couple $(1; 1)$ est une solution de l'inéquation $x_1 + 3x_2 \leq 8$ puisque : $1 + 3 \leq 8$ est vraie. Ce n'est pas la seule solution de cette inéquation ; les couples $(0; 0)$, $(1; 2)$, $(3; 1)$ sont également des solutions. L'ensemble de solutions de cette inéquation est un **demi-plan** dans le système d'axes Ox_1x_2 . La **frontière** de ce demi-plan est la droite $x_1 + 3x_2 = 8$.

Lorsque l'inéquation ne comporte qu'une inégalité stricte $<$ ou $>$ la frontière ne fait pas partie de l'ensemble-solutions de l'inéquation.

Cas $n=3$:

Lorsque l'inéquation comporte trois inconnues, la frontière est un plan. Ainsi la frontière de l'inéquation :

$$a_1x_1 + a_2x_2 + a_3x_3 \leq b$$

est un plan coupant les axes aux points représentant le triplet : $(b/a_1, 0, 0)$; $(0, b/a_2, 0)$; $(0, 0, b/a_3)$. Le plan, ensemble solutions d'une équation à 3 inconnues, divise l'espace en deux *demi-espaces*. L'un de ces demi-espaces en union avec la frontière forme l'ensemble-solution d'une inéquation linéaire à trois inconnues.

Cas $n>3$:

L'ensemble-solution d'une inéquation de la forme :

$$a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$$

est appelé **demi-espace fermé**.

Si l'inéquation est définie par une inégalité stricte ($<$), le demi-espace est dit **ouvert**. La frontière de ce demi-espace est l'ensemble solution de l'équation linéaire :

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b$$

Système d'inéquations linéaires:

On appelle **système de m inéquations linéaires à n inconnues** un système de la forme :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m \end{aligned}$$

Un n -uplet $(s_1; s_2; \dots; s_n)$ est **une solution d'un système de m inéquations à n inconnues** s'il est solution de chacune des inéquations du système.

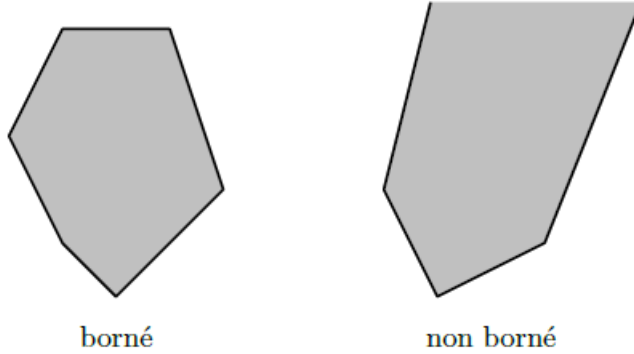
L'ensemble-solutions d'un système d'inéquations linéaires est l'intersection des ensembles-solutions de chacune des inéquations du système.

Théorème 4.1

- ❖ L'ensemble-solution d'une inéquation linéaire est un ensemble convexe.
- ❖ L'intersection de deux ou de plusieurs ensembles convexes est un ensemble convexe.
- ❖ L'ensemble-solutions d'un système d'inéquations linéaires est un ensemble convexe.

Remarque:

- ❖ L'ensemble-solutions d'un système d'inéquations linéaires à **2 (ou 3)** inconnues forme un polygone (ou un polyèdre) convexe.
- ❖ Dans le cas général de **système de m inéquations à n inconnues**, chaque contrainte inégalitaire est l'équation d'un demi-espace. Sa frontière est un hyperplan affine (i.e. un sous-espace affine de dimension 1) dans l'espace \mathbb{R}^n . Ainsi le domaine admissible d'un système d'inéquations est une intersection d'un nombre fini de demi-espaces. C'est donc un polytope convexe, ayant un nombre fini de sommets. Il peut être borné, ou non borné; ci-dessous un exemple de deux polygones convexes de \mathbb{R}^2 , l'un borné, l'autre non borné.

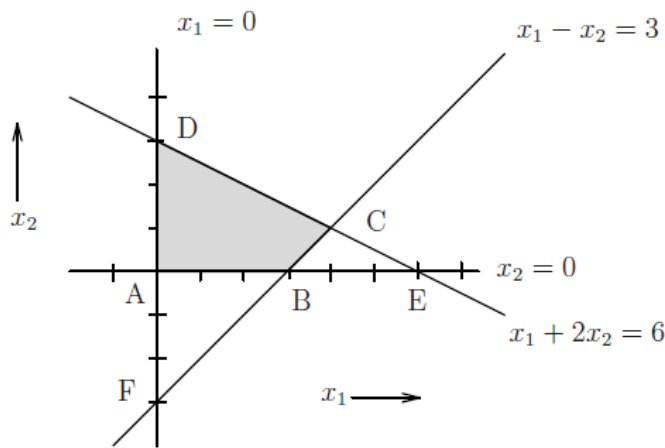


Exemple:

Soit le problème suivant:

$$(P) \quad \text{Maximiser } (x_1 + x_2) \text{ tel que } \begin{cases} x_1 + 2x_2 \leq 6 \\ x_1 - x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

L'ensemble des solutions du problème d'optimisation (P) qui représente le domaine des solutions réalisables est représenté par le polygone ci-dessous.



Programme Linéaire (forme canonique)

Un programme linéaire générique s'écrit sous **la forme canonique** comme suit:

$$\begin{cases} \max_x (c_1 x_1 + c_2 x_2 + \dots + c_n x_n) \\ \begin{pmatrix} a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n \\ \vdots \\ a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n \end{pmatrix} \leq \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \\ x_i \geq 0 \end{cases}$$

Ou, sous forme compact:

$$\begin{cases} \max_x \sum_{i=1}^n c_i x_i \\ \sum_{j=1}^n a_{ij} x_j \leq b_i, i = 1, m \\ x_i \geq 0 \end{cases}$$

$\sum_{i=1}^n c_i x_i$ est la fonction objective à maximiser sous les contraintes $\sum_{j=1}^n a_{ij} x_j \leq b_i, i = 1, m$
 Dont la forme matricielle est:

$$\begin{cases} \max_x c^t x \\ Ax \leq b \\ x_i \geq 0 \end{cases}$$

Avec

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \text{ et } A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}.$$

À noter que toute contrainte peut être transformée sous forme canonique. Deux cas peuvent alors se présenter.

1. Si la k^{ème} contrainte est de la forme :

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n \geq b_k \quad (i = 1, m)$$

Il suffit de multiplier par (-1) les deux membres pour obtenir

$$-a_{k1}x_1 - a_{k2}x_2 - \dots - a_{kn}x_n \leq -b_k \quad (i = 1, m)$$

2. Si la k^{ème} contrainte est de la forme :

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n = b_k \quad (i = 1, m)$$

On peut transformer cette équation en deux inéquations :

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n \leq b_k \quad (i = 1, m)$$

Et

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n \geq b_k \quad (i = 1, m)$$

Puis transformer la dernière en:

$$-a_{k1}x_1 - a_{k2}x_2 - \dots - a_{kn}x_n \leq -b_k \quad (i = 1, m)$$

Pour obtenir finalement:

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n \leq b_k \quad (i = 1, m)$$

Et

$$-a_{k1}x_1 - a_{k2}x_2 - \dots - a_{kn}x_n \leq -b_k \quad (i = 1, m)$$

Puisque les contraintes d'un programme linéaire ($Ax \leq b$) forment un système d'inéquations linéaires **dont le domaine admissible** est une intersection d'un nombre fini de demi-espaces. C'est aussi donc un polytope convexe, ayant un nombre fini de sommets. Il peut être borné, ou non borné;

Sachant qu'un domaine fermé, lorsqu'il est de plus borné c'est un domaine compact de \mathbb{R}^n . Or pour une application f linéaire et continue sur \mathbb{R}^n , si son domaine est borné, elle a forcément un minimum ainsi qu'un maximum. cela a pour conséquence que si un extremum existe, il est atteint sur l'un des sommets du domaine polytope (éventuellement sur tous les points d'une de ses faces, et en particulier sur l'un des sommets aussi).

Ces constatations sont résumées dans le théorème suivant :

Théorème 4.2 (Programmation linéaire)

En programmation linéaire, le domaine admissible, s'il est ni vide ni tout \mathbb{R}^n , est un polytope convexe ayant un nombre fini de sommets, qui peut être borné ou non borné. Si un extremum existe alors il est atteint sur l'un des sommets du polytope. Un point dans l'intérieur du domaine n'est jamais extrémal si $f \neq 0$. Lorsque le polytope est borné, f y prend un minimum ainsi qu'un maximum.

Programme Linéaire (forme standard ou normale)

La forme canonique est une forme très restrictive, si de plus, elle vérifie les deux conditions de positivité sur b_i et c_i , on obtient un problème de programmation linéaire sous forme normale comme suit:

$$\begin{cases} \max_x c^t x \\ Ax \leq b \\ x_i \geq 0 \\ b_i \geq 0 \\ c_i \geq 0 \end{cases}$$

On verra cependant par la suite comment ramener tout problème de programmation linéaire à un problème équivalent écrit sous forme normale.

Remarque 1:

D'une manière générale, on peut espérer imposer une contrainte $Ax \leq b$ ou $Ax \geq b$. Cette contrainte peut être transformée en une égalité en introduisant des variables artificielles appelées variables d'écarts. On écrit:

$$Ax + z = b, \quad z \in R^m \text{ et } z_i \geq 0$$

Il est souvent mathématiquement convenable de transformer toutes les inégalités en égalités.

On aura toujours la forme $Ax = b$ mais avec une matrice A et un vecteurs x des variables tout deux augmentées.

Nous nous intéressons dans ce chapitre uniquement aux systèmes d'équations linéaires avec un nombre n de variables supérieur au nombre m d'équations. Sachant que dans le cas de contraintes d'égalité linéaire ($Ax = b$ avec $A \in R^{m \times n}$), il y aura trois possibilités:

1. Aucune solution ;
2. Une solution unique ;
3. Une infinité de solutions.

Le second cas, avec une et une seule solution, ne peut survenir que si $n = m$, et si la matrice A est inversible, ce qui revient à exiger que nous avons éliminé au préalable toute équation pouvant s'écrire comme combinaison linéaire des autres équations. Si nous n'avons qu'une seule solution admissible, elle est forcément optimale, par conséquent, nous ignorerons ce cas dans notre étude, nous prendrons donc n strictement supérieur à m .

Plus précisément, pour $Ax = b$ avec $A \in R^{m \times n}$ et $m < n$. Alors on a n variables et m équations. On peut choisir (n-m) variables quelconques, les mettre à zéro; on aura une unique solution de $Ax = b$ pour les variables restantes (à condition que toutes les colonnes de la nouvelle matrice des variables restantes $A \in R^{m \times m}$ soient indépendantes c.à.d. son rang est m).

Caractérisation des solutions:

Nous avons montré comment un problème de programmation linéaire pouvait toujours se présenter sous forme standard :

$$(P) \begin{cases} \max_x c^t x \\ Ax = b \\ x_i \geq 0 \end{cases}$$

Dans cette formulation, le vecteur x contient toutes les variables, y compris les variables d'écart ; il s'agit d'un vecteur colonne d'ordre $(n \times 1)$. Les deux vecteurs b et c sont des vecteurs colonnes

d'ordres respectivement de $(n \times 1)$ et $(m \times 1)$. Quant à la matrice A , d'ordre $(m \times n)$, il s'agit de la matrice des coefficients des contraintes transformées.

On appelle **solution** d'un problème de programmation linéaire tout vecteur x qui satisfait les contraintes $(Ax = b)$. Une solution est appelée solution **réalisable** si elle vérifie les contraintes de non-négativité $(x_i \geq 0)$. Dans le cas contraire, on dit que la solution n'est pas réalisable.

Une solution réalisable est une solution **optimale** s'il n'existe pas d'autres solutions réalisables qui fournissent une plus grande valeur de la fonction objective. À noter que dans un problème possédant des solutions réalisables, il se peut que la valeur optimale de la fonction objective soit infinie. Dans ce cas, on parle de solution optimale infinie

Définition:

- Une solution **basique** du système $Ax = b$ est une solution avec au moins $(n-m)$ variables nulles.
- Une solution basique est **non-dégénéré** si exactement $(n-m)$ variables sont nulles.
- Le choix des m variables non-nulles est appelé base, les variables de base sont les variables basiques, les autres sont les non-basiques.
- Si une solution basique x satisfait $x \geq 0$, elle est une solution basique réalisable.

Supposition:

La matrice A $(m \times n)$ possède les propriétés suivantes:

- Rang $(A)=m$ où m est le nombre de contraintes.
- Toute combinaison de m colonnes de A est linéairement indépendante.
- Si x est une solution basique réalisable de (P) , alors x possède $(n-m)$ éléments non nulles (non-dégénérescence).

Théorème 4.3

Les solutions basiques réalisables d'un programme linéaire sont les points extrêmes de l'ensemble des solutions réalisables. Si un programme linéaire a une solution finie, alors il y a une solution basique réalisable optimale.

La supposition $\text{rang}(A)=m$, permet de construire à partir de la matrice A , une sous-matrice B $(m \times m)$ non-singulière. Cette matrice B peut être formée par n'importe quel ensemble de m colonnes linéairement indépendantes de A . Les colonnes de B seront notées b_1, b_2, \dots, b_m (à ne pas confondre avec le second membre b). La matrice B est appelée matrice de base puisqu'elle est formée de m vecteurs linéairement indépendants.

Sans perte de généralité, on peut supposer que les colonnes de A ont été ordonnées de manière à pouvoir écrire A sous la forme $A = (B, N)$, avec B de dimension $(m \times m)$ la matrice de base et N de dimension $(m \times (n-m))$ contient les colonnes de A qui n'appartiennent pas à B . Le vecteur x peut être partitionné de façon analogue en posant $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$. Les variables x_B sont appelés variables de base et les variables x_N les variables hors base. De même le vecteur c peut lui aussi être partitionné de la même manière en $c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}$.

Notre programme linéaire (P) peut donc être reformulé de la manière suivante :

$$(P) \begin{cases} \max_x c^t x \\ Ax = b \\ x_i \geq 0 \end{cases} \equiv \begin{cases} \max_x (c_B x_B + c_N x_N) \\ Bx_B + Nx_N = b \\ x_B, x_N \geq 0 \end{cases}$$

De la contrainte, on peut tirer:

$$Bx_B = b - Nx_N$$

Puisque la matrice B est non singulière donc inversible, on a:

$$x_B = B^{-1}b - B^{-1}Nx_N$$

Lorsque toutes les variables hors base sont nulles, $x_N = 0$, on aura $x_B = B^{-1}b$

On appelle solution de base la solution :

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$$

Lorsque $x_B = B^{-1}b \geq 0$ et $x_N = 0$, on parle de solution de base réalisable.

Exemple :

Soit le problème de programmation linéaire suivant :

$$\text{Maximiser } (3x_1 + 5x_2 + x_3) \text{ tel que } \begin{cases} x_1 + 2x_2 - x_3 \leq 16 \\ 3x_1 - 4x_2 \leq 20 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Ce problème peut se mettre sous forme standard en introduisant les variables d'écart x_4 et x_5 :

$$\text{Maximiser } (3x_1 + 5x_2 + x_3 + 0x_4 + 0x_5) \text{ tel que } \begin{cases} x_1 + 2x_2 - x_3 + x_4 = 16 \\ 3x_1 - 4x_2 + x_5 = 20 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Sous forme matricielle, on obtient donc :

$$\begin{cases} \max_x c^t x \\ Ax = b \\ x_i \geq 0 \end{cases}, \quad A = \begin{pmatrix} 1 & 2 & -1 & 1 & 0 \\ 3 & -4 & 0 & 0 & 1 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}, \quad b = \begin{pmatrix} 16 \\ 20 \end{pmatrix} \text{ et } c = (3 \quad 5 \quad 1 \quad 0 \quad 0)$$

Formons à partir de A une matrice de base B en prenant par exemple les colonnes 2 et 4, dans cet ordre :

$$B = \begin{pmatrix} 2 & 1 \\ -4 & 0 \end{pmatrix}$$

Il s'agit bien d'une base puisque B est non singulière car $\text{Det}(B) = 4 \neq 0$.

À cette matrice de base correspond une solution de base donnée par :

$$x_B = B^{-1}b = \begin{pmatrix} 0 & -1/4 \\ 1 & 1/2 \end{pmatrix} \begin{pmatrix} 16 \\ 20 \end{pmatrix} = \begin{pmatrix} -5 \\ 26 \end{pmatrix}$$

Les autres variables étant nulles, $x_1 = x_3 = x_5 = 0$. Cette solution de base n'est pas réalisable pour la simple raison que $x_B = \begin{pmatrix} -5 \\ 26 \end{pmatrix}$ viole la contrainte de non-négativité.

Dans cet exemple, il est très facile de trouver une base qui fournisse une solution réalisable de base. En effet les colonnes a_4 et a_5 forment une base qui est l'identité :

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Qui conduit à:

$$x_B = B^{-1}b = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 16 \\ 20 \end{pmatrix} = \begin{pmatrix} 16 \\ 20 \end{pmatrix}$$

Comme b doit être non-négatif lorsque le problème est présenté sous sa forme standard, la solution est une solution réalisable de base. Les autres variables étant nulles, $x_1 = x_2 = x_3 = 0$ et la fonction de coût sera: $0.16+0.20=0$.

Recherche de solution optimale:

En démarrant du principe que tout point extrême est une solution de base réalisable, on admet donc que la solution optimale est effectivement présente sur la frontière du simplexe, on peut imaginer qu'au lieu d'explorer tout l'espace continu du simplexe, on puisse se contenter de rechercher l'optimum sur ces points extrêmes ou de manière équivalente dans les solutions de base réalisables. À la solution réalisable de base initiale **correspond** une valeur de la fonction objective, **le but** est d'améliorer la valeur de la fonction objective en l'évaluant en un point extrême adjacent. Par exemple, **la fameuse méthode du simplexe** consiste à **se déplacer le long** d'une arête de la région réalisable étape par étape d'un point extrême donné à un point extrême voisin, jusqu'à ce que l'on arrive à un **point extrême optimal**. De tous les sommets adjacents, on choisit celui qui donne le plus grand accroissement de la fonction objectif (dans le cas d'une maximisation). À chaque point extrême, la méthode du simplexe nous dit s'il est optimal et si ce n'est pas le cas quel sera le prochain point extrême.

Si, à un certain moment, le sommet choisi possède une arête qui conduit à l'infini et si la fonction objective peut être améliorée en se déplaçant le long de cette arête, la méthode du simplexe nous informe qu'il y a une solution infinie.

Il a été dit plus haut que la méthode du simplexe commençait avec un point extrême donné. Le problème consiste alors à trouver un point extrême initial. Pour cela, étant donné une base B , on trouve un point extrême adjacent (nouvelle solution réalisable de base) en échangeant l'une des colonnes de la base (b_i) contre une colonne a_j qui n'est pas dans la base. Cependant, en faisant cet échange, il faut s'assurer que la solution de base reste réalisable et que la valeur de la fonction objectif augmente (ou du moins ne diminue pas).

Il y a donc deux règles à suivre pour cet échange. **La première** consiste à déterminer quelle colonne a_j de A (à laquelle correspond une variable x_j) doit entrer dans la base pour améliorer la fonction objective. **La seconde** consiste à sélectionner l'une des colonnes b_i qui doit quitter la base de manière à ce que la nouvelle solution de base reste réalisable. Pour cela, il faut développer des critères d'entrée et de sortie de la base pour obtenir une nouvelle solution réalisable de base qui améliore la valeur de la fonction objective. **Il restera ensuite à déterminer** si la nouvelle solution réalisable de base est optimale ou non.

De cette manière, on finit par rechercher l'optimum sur un ensemble fini de points qui sont les sommets du simplexe. Chacun de ces points forment une base du simplexe, et peuvent être trouvé **par manipulation du système linéaire d'inéquations ce qui constitue l'algorithme de la méthode du simplexe** développé en 1947 par George Dantzig, la méthode du simplexe reste d'actualité pour résoudre des problèmes de grande taille. Il s'agit d'une méthode algébrique basée sur la résolution de systèmes d'équations linéaires

La méthode simplexe

La procédure algorithmique décrite ci-dessus pour résoudre un programme linéaire écrit sous forme standard et arriver à l'optimum peut être formalisé et systématiser par manipulation de tableaux. En effet, au lieu d'écrire à chaque fois toutes les équations, plutôt on joue sur la distribution des coefficients dans un tableau. C'est le principe de la méthode simplexe du tableau qu'on va décrire ci-dessous, les gens intéressés peuvent consulter les développements mathématiques de la méthode algébrique complète en littérature. Même si la méthode du simplexe du tableau se prête mal à l'implémentation informatique, c'est un outil très simple et puissant qui résume toutes les étapes algorithmiques de la méthode algébrique.

La méthode simplexe du tableau

Cette méthode est **basée** sur la formulation standard ou **normale du programme** linéaire rappelée ci-dessous:

$$\begin{cases} \max c^t x \\ Ax \leq b \\ x_i \geq 0 \\ b_i \geq 0 \\ c_i \geq 0 \end{cases}$$

Une fois le problème d'optimisation est écrit sous cette forme, on procède comme suit:

1) on effectue les étapes suivantes

a. On change chacune des m contraintes inégalitaires en une contrainte égalitaire en introduisant une **variable d'écart** notée s_i ($i = 1; \dots; m$) pour arriver enfin à:

$$Ax + s = b, \quad s \in R^m \text{ et } s_i \geq 0$$

b. En cas où une (ou certaines) variable x_i ne vérifie pas la contrainte de positivité c.à.d. si $x_i < 0$, il suffit de **poser** $x_i = x_i^+ - x_i^-$ **avec** $x_i^+, x_i^- > 0$.

c. En cas de présence de contrainte égalitaire c.à.d. si pour un certain (i,j) on a $a_{ij} x_j = b_i$, il faut l'insérer dans le tableau telle qu'elle est **sans ajouter** de variable d'écart.

d. si un certain $c_i < 0$, il suffit de poser $x_i = 0$ dans la matrice de la méthode du simplexe cela revient à **supprimer** la colonne correspondante.

e. si un certain $b_i < 0$, on **change** la contrainte inégalitaire en une égalitaire en insérant un variable supplémentaire $p_i \geq 0$ comme suit: si $b_i < 0$

$$\Rightarrow a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq -|b_i| \Rightarrow -a_{i1}x_1 - a_{i2}x_2 - \dots - a_{in}x_n - p_i = |b_i|$$

2) on construit la matrice suivante:

x_1	...	x_n	s_1	...	s_m	b } première ligne
a_{11}	...	a_{1n}	1	0	0	$\left. \begin{matrix} b_1 \\ \vdots \\ b_m \end{matrix} \right\}$ partie centrale
\vdots	\ddots	\vdots	0	\ddots	0	
a_{m1}	...	a_{mn}	0	0	1	
$\underbrace{c_1 \quad \dots \quad c_n}_{c^t}$			0	...	0	f - 0 } ligne résultat

En notant la ligne résultat de la partie gauche par:

$$r = (c_1 \quad \dots \quad c_n \quad 0 \quad \dots \quad 0) = (r_1 \quad \dots \quad r_{n+m})$$

La première ligne est optionnelle et purement nominative ; la dernière ligne s'appelle ligne résultat. Le trait vertical (en gras) symbolise l'égalité ; il sépare la partie gauche de la colonne droite.

3) travaillez sur ce tableau en appliquant les transformations suivantes:

- choisir une variable à entrer dans la base: piquer j tel que $r_j > 0$ (la variable correspondante x_j entre dans la base), s'il y a plusieurs choisissez la **plus grande**.
- choisir une variable qui doit quitter la base: choisir la ligne pivot i dans la colonne pivot j, c'est un élément (i,j) de la partie centrale gauche que l'on note α_{ij} choisi de façon à ce **que** $\frac{b_i}{\alpha_{ij}}$ **soit** ≥ 0 **et minimal**. Si $\alpha_{ij} \leq 0$ pour tout i, alors le problème est non borné et la fonction

objective peut être augmenté sans limite **donc arrêter pas** de solution. S'il y a plus d'un seul i qui minimise $\frac{b_i}{\alpha_{ij}}$, alors le problème a une solution basique réalisable **dégénérée**; on peut choisir **un quelconque et continuer**.

- Une fois la ligne pivot i obtenue que l'on note (L), multiplier toute la ligne pivot par $\frac{1}{\alpha_{ij}}$ puis ajouter et soustraire autant de fois que nécessaire de lignes pivot au autres lignes de façon à **annuler tous les termes** de la colonne pivot j **autre que celui** de la ligne pivot i .
- Si tous les $r_j \leq 0, j \geq 1$, **arrêter** alors, la solution actuelle est optimale.
- Enfin lorsqu'on s'arrête, il faut **barrer de la matrice toutes** les colonnes j dont $r_j < 0$ et **poser** les variables correspondantes à **zéro**.
- **Résoudre** le système linéaire de la **partie centrale** pour trouver les valeurs x_i qui maximisent la fonction objective. La **valeur maximale** de la fonction objective f_{max} **se lit** directement dans la case **en bas à droite**.

Exemple d'application:

Soit à résoudre le problème d'optimisation (linéaire) suivant:

$$\begin{cases} \max_x f(x) = 150x_1 + 450x_2 \\ x_1 \leq 120 \\ x_2 \leq 70 \\ x_1 + x_2 \leq 140 \\ x_1 + 2x_2 \leq 180 \\ x_1, x_2 \geq 0 \end{cases}$$

Matrice initiale

x	y	s_1	s_2	s_3	s_4	
1	0	1	0	0	0	120
0	1	0	1	0	0	70
1	1	0	0	1	0	140
1	2	0	0	0	1	180
150	450	0	0	0	0	$f - 0$



Sélection de la colonne pivot et de la ligne pivot

x	y	s_1	s_2	s_3	s_4	
1	0	1	0	0	0	120
0	1	0	1	0	0	70
1	1	0	0	1	0	140
1	2	0	0	0	1	180
150	450	0	0	0	0	$f - 0$



Opérations pour annuler les termes de colonnes pivot autre que celui de la ligne pivot

x	y	s_1	s_2	s_3	s_4		
1	0	1	0	0	0	120	
0	1	0	1	0	0	70	(L)
1	1	0	0	1	0	140	-(L)
1	2	0	0	0	1	180	-2(L)
150	450	0	0	0	0	$f - 0$	-450(L)



Premier tableau résultat

x	y	s_1	s_2	s_3	s_4		
1	0	1	0	0	0	120	
0	1	0	1	0	0	70	
1	0	0	-1	1	0	70	
1	0	0	-1	0	1	40	
150	0	0	-450	0	0	$f - 31500$	



Mêmes opérations pour arriver au deuxième tableau résultat suivant

x	y	s_1	s_2	s_3	s_4		
0	0	1	1	0	-1	80	
0	1	0	1	0	0	70	
0	0	0	0	1	-1	30	
1	0	0	-1	0	1	40	
0	0	0	-300	0	-150	$f - 37500$	

Tous les $r_j \leq 0, j \geq 1$; alors **fmax=37500** et la partie centrale conduit à $x_1 = 40$ et $x_2 = 70$

Exemple 1:

Un artisan fabrique deux types de sacs à main en cuir pour femmes, il dispose de 40 m² de cuir chaque semaine alors qu'il peut travailler au maximum 60 heures par semaine.

Si chaque type de sac demande un 1m² de cuir et une heure de travail pour le type standard mais deux heures sont nécessaires pour finaliser un sac luxe.

Comment doit-il travailler pour maximiser son profit hebdomadaire si le sac luxe lui rapporte 4U et le sac standard lui rapporte 3U?****Utilisez la méthode du Simplexe

Solution:

Si nous appelons x_1 le nombre de sacs standard et x_2 le nombre de sacs luxes, la fonction de coût à maximiser est donnée par:

$$f(x) = f(x_1, x_2) = 3x_1 + 4x_2$$

Les contraintes sont:

$$\begin{aligned} x_1 + x_2 &\leq 40 \\ x_1 + 2x_2 &\leq 60 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

La forme standard de ce problème est la suivante:

$$\text{Max } f(x) = C^t x$$

$$\text{tel que } \begin{cases} Ax \leq b \\ x \geq 0 \end{cases} \quad \text{Avec} \quad C = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, b = \begin{pmatrix} 40 \\ 60 \end{pmatrix}$$

x_1	x_2	s_1	s_2		
1	1	1	0	40	-L/2
1	(2)	0	1	60	(L)
3	<u>4</u>	0	0	f-0	-2L



x_1	x_2	s_1	s_2		
1/2	0	1	-1/2	10	(L)
1	2	0	1	60	(-2L)
<u>1</u>	0	0	-2	f-120	(-2L)



x_1	x_2	s_1	s_2		
(1/2)	0	1	-1/2	10	
0	2	-2	2	40	
0	0	-2	-1	f-140	

Ce qui conduit à $\max f(x) = 140$ en $x_1 = 20$ et $x_2 = 20$.

Exemple 2:

Un fournisseur de fruits/légumes a promis à sa clientèle qu'au moins 25% de sa marchandise serait d'origine "Bio". Ayant une commande de 18 tonnes de marchandises, il a sélectionné 3 agriculteurs pour s'approvisionner; le taux de "biovicité" des marchandises et la marge bénéficiaire de chacun de ces fournisseurs en gros sont résumés dans le tableau ci-dessous.

	% bio de sa marchandise	Quantité de marchandise proposée en tonnes	Marge bénéficiaire en kilo Dollars par tonne
Agriculteur 1	10%	25	900

Agriculteur 2	46%	6	700
Agriculteur 3	100%	4	500

Comment ce fournisseur doit-il s'approvisionner pour maximiser son bénéfice? Utilisez la méthode du simplexe pour résoudre ce problème.

Solution:

Il faut commencer à trouver la fonction objective à optimiser, cela en cherchant à maximiser le bénéfice du fournisseur. Si nous appelons q_1 la quantité (en tonnes) achetée de l'agriculteur 1, q_2 la quantité achetée de l'agriculteur 2 et q_3 celle achetée de l'agriculteur 3. La fonction objective (en kilo Dollars) sera de la forme: $f = 900q_1 + 700q_2 + 500q_3$.

Puis voir

- Qu'on ne peut pas vendre plus de 18 tonnes $\Rightarrow q_1 + q_2 + q_3 \leq 18$.
- Or les quantités proposées par les agriculteurs sont limitées $\Rightarrow q_1 \leq 25, q_2 \leq 6$ et $q_3 \leq 4$
- La quantité bio de toute la marchandise achetée est égale à: $(0.1 \times q_1 + 0.46 \times q_2 + q_3)$ qui doit être supérieur ou égale à $0.25 \times q$ où $q = q_1 + q_2 + q_3$ est quantité totale achetée, on doit avoir donc

$$(0.1 q_1 + 0.46 q_2 + q_3) \geq 0.25(q_1 + q_2 + q_3)$$

Ce qui conduit à:

$$0.15 q_1 - 0.21 q_2 - 0.75 q_3 \leq 0$$

Et le problème d'optimisation s'écrit:

$$\left\{ \begin{array}{l} \max_x f(x) = 900q_1 + 700q_2 + 500q_3 \\ q_1 \leq 25 \\ q_2 \leq 6 \\ q_3 \leq 4 \\ q_1 + q_2 + q_3 \leq 18 \\ 0.15 q_1 - 0.21 q_2 - 0.75 q_3 \leq 0 \\ q_1, q_2, q_3 \geq 0 \end{array} \right.$$

Bien sur les quantités sont des chiffres positifs:

$q_1, q_2, q_3 \geq 0$ ce qui fait $q_1 + q_2 + q_3 \leq 18 \Rightarrow q_1 \leq 25$ cette dernière contrainte peut être supprimé.

L'application de la méthode du simplexe de tableau conduit aux valeurs optimales $q_1 = 15, q_3 = 3$ et $q_2 = 0$. Ainsi le fournisseur doit acheter 15 tonnes de l'agriculteur 1 et 3 tonnes de l'agriculteur 3, et le bénéfice maximal est calculé selon l'expression de la fonction objective: $f_{\max} = (900 \times 15 + 700 \times 0 + 500 \times 3)$ kilodollars = 15 millions de dollars.

Dualité minimum/maximum

La méthode simplexe du tableau décrite ci-dessus est une méthode de calcul du maximum de la fonction de coût d'un programme linéaire. Si on veut l'appliquer à la minimisation, il faut transformer le problème d'optimisation en un problème de minimisation en s'aidant du théorème de dualité ci-dessous.

Un problème de minimisation s'écrit sous forme canonique :

$$\begin{cases} \min_y b^t y \\ A^t y \geq c \\ y \geq 0 \end{cases}$$

Si de plus $b, c \geq 0$, on aura la forme standard ou normale.

Théorème 4.4

Tout problème de minimisation linéaire sous forme normale est équivalent à un problème de maximisation linéaire sous forme normale dans le sens suivant :

$$\begin{cases} \min_y g(y) = b^t y \\ A^t y \geq c \\ y \geq 0 \\ b, c \geq 0 \end{cases} \equiv \begin{cases} \max_x f(x) = c^t x \\ Ax \leq b \\ x \geq 0 \\ b, c \geq 0 \end{cases}$$

$g_{\min} = f_{\max}$, le minimum de g a pour coordonnées les opposés des valeurs dans la ligne résultat correspondant aux variables d'écart du **problème de maximisation**.

Exemple 1:

Soit le problème de minimisation suivant:

$$\begin{cases} \min_y g(y) = 2y_1 + 8y_2 \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 3 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \geq \begin{pmatrix} 100 \\ 100 \\ 500 \\ 900 \\ 1200 \end{pmatrix} \\ y_1 \geq 0 \\ y_2 \geq 0 \end{cases}$$

D'après la forme standard de minimisation, on identifie:

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 & 3 \\ 0 & 1 & 1 & 3 & 2 \end{bmatrix}, c = \begin{pmatrix} 100 \\ 100 \\ 500 \\ 900 \\ 1200 \end{pmatrix}, b = \begin{pmatrix} 2 \\ 8 \end{pmatrix}$$

On aura l'équivalent sous forme de problème de maximisation suivant:

$$\begin{cases} \max_x f(x) = 100x_1 + 100x_2 + 500x_3 + 900x_4 + 1200x_5 \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 3 \\ 0 & 1 & 1 & 3 & 2 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \leq \begin{pmatrix} 2 \\ 8 \end{pmatrix} \\ x_i \geq 0 \end{cases}$$

Que l'on peut résoudre par la méthode de simplexe du tableau (sans représenter la première ligne $(x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ s_1 \ s_2 \ |b)$ comme suit:

$$\begin{array}{cccc|cc|c} 1 & 0 & 1 & 1 & \boxed{3} & 1 & 0 & 2 \\ 0 & 1 & 1 & 3 & 2 & 0 & 1 & 8 \\ \hline 100 & 100 & 500 & 900 & 1200 & 0 & 0 & f - 0 \end{array}$$



$$\begin{array}{cccc|cc|c} 1 & 0 & 1 & \boxed{1} & 3 & 1 & 0 & 2 \\ -2/3 & 1 & 1/3 & 7/3 & 0 & -2/3 & 1 & 20/3 \\ \hline -300 & 100 & 100 & 500 & 0 & -400 & 0 & f - 800 \end{array}$$



$$\begin{array}{cccc|cc|c} 1 & 0 & 1 & 1 & 3 & 1 & 0 & 2 \\ -3 & \boxed{1} & -2 & 0 & -7 & -3 & 1 & 2 \\ \hline -800 & 100 & -400 & 0 & -1500 & -900 & 0 & f - 1800 \end{array}$$



$$\begin{array}{cccc|cc|c} 1 & 0 & 1 & 1 & 3 & 1 & 0 & 2 \\ -3 & 1 & -2 & 0 & -7 & -3 & 1 & 2 \\ \hline -900 & 0 & -200 & 0 & -800 & -\boxed{600} & -\boxed{100} & f - \boxed{2000} \end{array}$$

Ainsi $g_{\min} = 2000 = g(600, 100) = 2 \times 600 + 8 \times 100$.

Exemple 2:

On peut revenir à notre exemple d'introduction du problème de transport:

On considère un problème de ravitaillement très utile en pratique; il fait partie d'une large classe de problème (largement étudiée que ce soit en optimisation continue ou en optimisation combinatoire, et que l'on sait résoudre efficacement).

On souhaite approvisionner en carburant 3 sites à partir de 2 dépôts de capacité limitée.

L'acheminement en carburant d'un dépôt à un site a un coût unitaire. Le tableau suivant résume chacun de ces coûts ainsi que la demande de chaque site, et le stock disponible dans chaque dépôt.

	site 1	site 2	site 3	diponibilité
dépôt 1	10	12	9	300
dépôt 2	11	11	10	450
demande	200	250	250	

Solution:

Notons $x_{i,j}$ $i=1, 2$ et $j=1, 2, 3$ la quantité de carburant (en unité de volume) acheminé du dépôt i au site j . Le coût d'acheminement est donné par la fonction de coût suivante :

$$f(x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}) = 10x_{11} + 12x_{12} + 9x_{13} + 11x_{21} + 11x_{22} + 10x_{23}$$

qu'il s'agit de **minimiser** la fonction de coût, sous les contraintes :

- ❖ Contraintes égalitaires provenant de la demande :

$$\begin{cases} x_{11} + x_{21} = 200 \\ x_{12} + x_{22} = 250 \\ x_{13} + x_{23} = 250 \end{cases}$$

❖ Contraintes inégalitaires provenant du stock disponible :

$$\begin{cases} x_{11} + x_{12} + x_{13} \leq 300 \\ x_{21} + x_{22} + x_{23} \leq 450 \end{cases}$$

❖ Contraintes de signe :

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23} > 0.$$

Il s'agit là d'un problème de minimisation qu'il faut transformer puis lui appliquer la méthode de simplexe de tableau.

Remarque générale:

On remarque qu'en pratique, la plus grande difficulté qui se pose pour résoudre un problème de programmation linéaire est la formulation mathématique sous forme canonique ou standard du problème pratique, cette étape est la plus importante. La suite du travail est juste une manipulation de tableaux.

Annexe

Inéquations Matricielles Linéaires (LMI)

Définition: (Forme quadratique)

Un ensemble quadratique est de la forme suivante :

$$S = \{x \in R^n | x^t Q x + 2h^t x + k \leq 0\}$$

$$= \left\{ x \in R^n \mid \begin{pmatrix} x \\ 1 \end{pmatrix}^t \begin{pmatrix} Q & h \\ h^t & k \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} \leq 0 \right\}$$

$Q \in R^n \times R^n$, $h \in R^n$ et $k \in R$

Définition: (Hyperplan)

Un demi-plan (hyperplan) dans R^n est un ensemble de la forme :

$$H = \{x \in R^n | a^t x = b\}, \quad a \in R^n, a \neq 0_n, b \in R$$

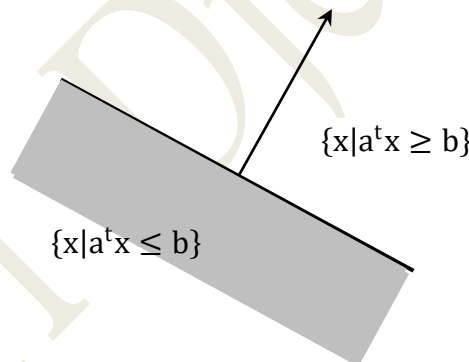
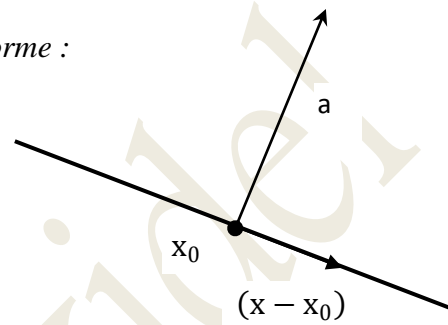
Ou équivalamment :

$$H = \{x \in R^n | a^t (x - x_0) = 0\}, \quad b = a^t x_0$$

Où a est le vecteur normal au plan.

Un hyperplan est un ensemble affine (dimension $(n-1)$) et convexe, il divise l'espace R^n en deux demi-espaces ouverts.

$$H^- = \{x \in R^n | a^t x - b < 0\} \quad \text{et} \quad H^+ = \{x \in R^n | a^t x - b > 0\}$$



L'intersection d'un nombre fini d'hyperplan est un ensemble affine :

$$A = \{x \in R^n | Ax = b\}, \quad A \in R^{m \times n}, b \in R^m$$

Définition: (Polyèdre)

L'intersection définie par un nombre fini de demi-espaces est un polyèdre (ensemble convexe)

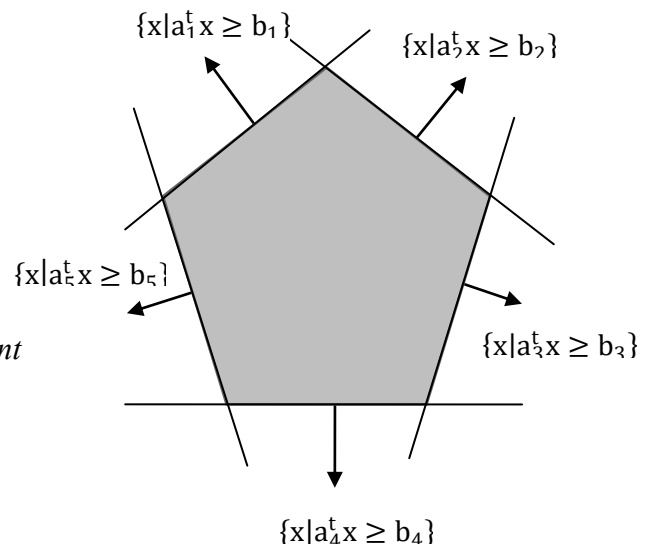
$$P = \{x \in R^n | a_i^t x - b_i < 0, \quad i = 1, 2, \dots, m\}$$

$$= \bigcap_{i=1}^m [a_i^t x - b_i < 0]$$

$$= \{x \in R^n | Ax - b < 0\}, \quad A \in R^{m \times n}, b \in R^m$$

D'autre part : $\bar{P} = \bigcap_{i=1}^m [a_i^t x - b_i \leq 0]$

Où $F = P \cap \{x \in R^n | a_i^t x = b_i\}$ est appelée face du polyèdre ; faces de dimension 0, 1 et $(n-1)$ sont appelées sommets, arrêtes et faces.



Définition: (Polytope)

Un polytope est un polyèdre fermé et borné où chaque point peut être représenté par une combinaison convexe de ses sommets. Il représente donc la coque de ses points sommets, cette représentation est appelée *v*-représentation qui a l'avantage d'être décrite par des hyperplans.

$$\mathcal{P} = \text{co}(\{v_1, v_2, \dots, v_{nv}\}),$$

Beaucoup d'algorithmes d'optimisation sont basés sur la représentation polytopique ou ellipsoïdale des régions d'espace.

Définition: (Ellipsoïde)

Un ellipsoïde peut être défini de différentes manières, nous retiendrons les trois formes ci-dessous où le passage entre ces deux représentations est possible.

Un ellipsoïde centré en x_c , est un ensemble décrit uniquement par x_c et la matrice P :

$$\varepsilon = \{x \in R^n | (x - x_c)^t P (x - x_c) \leq 1\}, \quad P^t = P > 0$$

Ou équivalamment, un ellipsoïde peut être décrit par une forme quadratique générale :

$$\varepsilon = \{x \in R^n | x^t P x + 2b^t x + c \leq 0\}, P^t = P > 0, b^t P^{-1} b - c > 0$$

$$\varepsilon = \left\{x \in R^n \mid \begin{pmatrix} x \\ 1 \end{pmatrix}^t \begin{pmatrix} P & b \\ b^t & c \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} \leq 0\right\}, P^t = P > 0, b^t P^{-1} b - c > 0$$

La dernière condition permet d'assurer que l'ellipsoïde ne se réduit pas à un point et ne soit pas vide. Cette représentation n'est pas unique, la multiplication par un même facteur positif de A , b et c tient toujours. En effet par identification, nous avons : $b = -Px_c$ et $c = x_c^t P x_c - 1$. Une autre manière de décrire un ellipsoïde est de considérer la déformation de la sphère unité par une matrice Q qui est l'inverse du facteur de Sholesky de la matrice définie positive P : $Q = P^{-\frac{1}{2}}$.

$$\varepsilon = \{x = Qy + x_c | y^t y \leq 1\}, \quad P^t = P > 0$$

Le volume de l'ellipsoïde est proportionnel au déterminant de Q . Si les λ_i sont les valeurs propres de P , l'ellipsoïde admet des demi-axes de longueurs $1/\sqrt{\lambda_i}$. Puisque le déterminant de P est égal au produit de ses valeurs propres et la trace de P est égale à leur somme, ces paramètres peuvent être utilisés pour maximiser ou minimiser la taille et par conséquent le volume de l'ellipsoïde.

$$\log \det P = \log \lambda_1 + \log \lambda_2 + \dots + \log \lambda_n.$$

$$\text{trace } P = \lambda_1 + \lambda_2 + \dots + \lambda_n$$

Nous rencontrons souvent des questions d'optimisations de type min et max de ces paramètres.

Ex1 :

Min $\log \det P$

$$\text{tel que } \begin{cases} P^t = P > 0 \\ F(P^{-1}) > 0 \end{cases}$$

Où $F(P^{-1})$ est une contrainte LMI en P^{-1} , et donc l'optimisation est convexe en P^{-1} .

Ex2 :

Min $\text{trace } P$

$$\text{tel que } \begin{cases} P^t = P > 0 \\ F(P) > 0 \end{cases}$$

Où $F(P)$ est une contrainte LMI en P , et nous avons le problème d'optimisation LMI classique à critère linéaire.

Ex3 :

Problème de minimisation de la plus grande valeur propre de P ou équivalamment la minimisation de la borne supérieur sur cette valeur.

$$\text{Min } \lambda \quad \text{tel que } \begin{cases} \lambda I \geq P \\ P^t = P > 0 \\ F(P) > 0 \end{cases}$$

Où F(P) est une contrainte LMI en P, et nous avons le problème d'optimisation LMI classique à critère linéaire.

Remarque :

En commande, on aura souvent besoin d'optimiser le volume de forme polyédrale fermée i.e. polytope, pour cette question, il n'y a pas d'expression analytique à optimiser. Le moyen le plus simple consiste à résoudre ce problème indirectement en optimisant le volume de l'ellipsoïde contenu/ contenant le polytope en question.

Par exemple, l'inclusion de l'ellipsoïde $\varepsilon = \{x = Qy + x_c | y^t y \leq 1\}$, dans le polyèdre $P = \{x \in R^n | a_i^t x - b_i < 0\}$ est équivalent à $\forall i = 1, 2, \dots, m, a_i^t(Qy + x_c) - b_i \leq 0$ pour tout vecteur y tel que $y^t y \leq 1$. Ce maximum est atteint pour i tel que $y = \frac{Qa_i}{\|Qa_i\|}$ ce qui conduit aux lemmes suivants :

Lemme A1 :

L'ellipsoïde $\varepsilon = \{x = Qy + x_c | y^t y \leq 1\}$,

Est inclus dans le polyèdre $P = \{x \in R^n | a_i^t x - b_i \leq 0\}$,

Si et seulement si $\|Qa_i\| + a_i^t x_c - b_i \leq 0$

L'application du complément de Schur conduit à :

$$\begin{bmatrix} (b_i - a_i^t x_c)P & a_i \\ a_i^t & (b_i - a_i^t x_c) \end{bmatrix} \geq 0, \quad \forall i = 1, 2, \dots, m \text{ et } P = Q^{-2}.$$

Cette inéquation est facilement solvable (LMI) si x_c est connu ou nul (ellipsoïde centré à l'origine).

Lemme A2 :

L'ellipsoïde centré à l'origine $\varepsilon = \{x | x^t P x \leq 1\}$

Est inclus dans le polyèdre $P = \{x \in R^n | a_i^t x - b_i \leq 0\}$,

Si et seulement si $\begin{bmatrix} P & a_i \\ a_i^t & b_i^2 \end{bmatrix} \geq 0, \quad \forall i = 1, 2, \dots, m$

Définition: (Cône)

Un cône elliptique de sommet x_c est un ensemble de la forme :

$$Co = \{x \in R^n | (x - x_c)^t P (x - x_c) \leq 0\}$$

Où $P^t = P \in R^n$ possède $(n-1)$ valeurs propres positives et une valeur propre négative.

Ou après changement de variable :

$$Co = \left\{ \begin{pmatrix} y \\ y_n \end{pmatrix} \in R^n \mid y^t \tilde{P} y \leq y_n^2 \right\}, \quad \tilde{P} \in R^{(n-1) \times (n-1)}, \quad \tilde{P}^t = \tilde{P} > 0, y_n \geq 0$$

On peut choisir $y_n = 1$ si on désire limiter le volume du cône et en utiliser la formule du volume de l'ellipsoïde.

Régions coniques :

- Si une région d'espace Ω_i contient l'origine et doit être délimitée par des demi-espaces, ils doivent être au moins au nombre de deux et elle peut être décrite par une forme quadratique comme suit :

$C_1^t x \geq 0$ et $C_2^t x \geq 0$ Elle peut être décrite par la forme quadratique suivante :

$$x^t Q_0 x \geq 0 \text{ avec } Q_0 = C_1 C_2^t + C_2 C_1^t$$

- Si une région d'espace Ω_i ne contient pas l'origine $C^t x + d \geq 0$, elle peut être décrite par la quadratique suivante :

$$z Q_1 z^t \geq 0 \text{ où } z = [x \ 1] \text{ et } Q_1 = [0_n \ C; \ C^t \ 2d].$$

- Si une région d'espace Ω_i est décrite par:

$$C_1^t x + d_1 \geq 0 \text{ et } C_2^t x + d_2 \geq 0$$

Alors il faut prendre en considération toutes les combinaisons des demi-espaces.

Inégalités matricielles linéaires (LMI) :

Une inéquation matricielle linéaire (ou LMI selon l'acronyme anglo-saxon) est une expression de la forme :

$$F(x) = F_0 + x_1 F_1 + \dots \dots x_n F_n > 0 \quad (1-1)$$

Où

- $x = (x_1, x_2 \dots \dots x_n)$ est un vecteur de n nombres réels appelés variables de décision.
- $F_0, F_1, \dots \dots F_n$ est un ensemble de matrices symétriques réelles (ou complexes), donc ($F_i \in D^{m \times m}$ (ou $H^{m \times m}$)).
- Le symbole d'inéquation $>$ signifie que $F(x)$ est définie positive i.e. les principaux mineurs de $F(x)$ sont positifs.

Donc $F: \mathbb{X} \rightarrow D^{m \times m}$ (ou $H^{m \times m}$) est une fonction affine de l'espace vectoriel \mathbb{X} dans $S^{m \times m}$ (ou $H^{m \times m}$).

Pour la fonction affine qui associe à chaque x , la fonction :

$$F(x) = F_0 + T(x) \quad (1-2)$$

Où T est une transformation linéaire, si \mathbb{X} est de dimension finie n de base $(e_1, e_2 \dots \dots e_n)$ donc chaque $x \in \mathbb{X}$ peut s'écrire: $x = \sum_{j=1}^n x_j e_j$, par suite on a :

$$T(x) = T\left(\sum_{j=1}^n x_j e_j\right) = \sum_{j=1}^n x_j F_j, \quad F_j = T(e_j). \quad (1-3)$$

Finalement la forme (1-2) est obtenue.

LMI en automatique :

Généralement en automatique l'expression (1-1) apparaît comme fonction de variables matricielles au lieu de variables de décision scalaires i.e. $\mathbb{X} = R^{m \times m}$, un bon exemple est l'équation de Lyapunov : $F(X) = A^t X + X A + Q < 0$ où A, Q sont des matrices $R^{m \times m}$ données et X est la matrice inconnue de dimension $m \times m$. cette LMI est un cas particulier de (1-2) (tout en prenant $F_0 = 0$ et $F(X) = -A^t X - X A - Q$) en considérant une base quelconque de \mathbb{X} $(e_1, e_2 \dots \dots e_n)$, tout X appartenant à \mathbb{X} est représenté par : $X = \sum_{j=1}^n x_j e_j$, ce qui donne :

$$F(X) = F\left(\sum_{j=1}^n x_j e_j\right) = F_0 + \sum_{j=1}^n x_j F_j. \quad (1-4)$$

Les coefficients x_j dans le développement de X , constituent les variables de décision, leur nombre n correspond à la dimension de X . Le nombre n est tel que $\text{Max}(n)=m^2$ selon la structure imposée à X , pour une matrice X symétrique de dimension $m \times m$, on aura $n = m(m + 1)/2$ éléments.

La LMI (1-4) est convexe en la contrainte x i.e. l'ensemble $\{X|F(X) > 0\}$ est convexe, elle peut prendre n'importe quelle forme spéciale et couvrir donc une grande variété de problèmes convexes.

Propriétés :

- Plusieurs LMI peuvent se transformer en une seule LMI, en effet, $F_1(x) > 0, F_2(x) > 0, \dots \dots F_k(x) > 0$ est équivalent à $\text{diag}(F_1(x), F_2(x), \dots \dots F_k(x)) > 0$.
- Transformation de congruence : pour une matrice M carrée et T non singulière, le produit T^tMT est appelé transformation de congruence de M . cette transformation ne change pas le nombre de valeurs propres négatives et positives par conséquent $M > 0$ si et seulement si $T^tMT > 0$.

Considérons une matrice $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$, M_{11} carrée et non singulière et la matrice

non singulière $T_1 = \begin{bmatrix} I & 0 \\ -M_{11}^{-1}M_{12}^t & I \end{bmatrix}$ ce qui conduit à :

$$M > 0 \Leftrightarrow \begin{bmatrix} M_{11} & 0 \\ 0 & M_{22} - M_{12}^t M_{11}^{-1} M_{12} \end{bmatrix} > 0 \tag{1-5}$$

De même en considérant $T_2 = \begin{bmatrix} I & -M_{22}^{-1}M_{21}^t \\ 0 & I \end{bmatrix}$ on obtient :

$$M > 0 \Leftrightarrow \begin{bmatrix} M_{11} - M_{21}^t M_{22}^{-1} M_{21} & 0 \\ 0 & M_{22} \end{bmatrix} > 0 \tag{1-6}$$

(1-5) et (1-6) constituent bien le puissant outil du complément de Schur qui permet de linéariser certaines inéquations matricielles non linéaires (NMI).

• **LMI non strictes :**

Les problèmes LMI peuvent être constitués par des LMI stricts et/ou de LMI non stricts. Généralement, lorsqu'il y a solution aux LMI stricts (celles prises en charge par le toolbox LMI du Matlab), il y a aussi solution aux LMI non stricts. De plus, lorsqu'il y a optimisation de certain critère supplémentaire, les deux valeurs optimums coïncident. Cependant, certaines LMI non strictes n'auront pas de solutions lorsqu'elles sont remplacées par des LMI strictes :

Exemples :

$$\begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} \geq 0 \text{ a une solution (} P = 0 \text{) alors que } \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} > 0 \text{ n'a pas de solution,}$$

De même pour :

$$\begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} \geq 0 \text{ a une solution pour (} P > 0 \text{) alors que } \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} > 0 \text{ n'a pas de solution.}$$

Certaines LMI non strictes ou LMI multiple (p.ex. : LMI strictes avec contrainte additionnelle d'égalité) peuvent être transformées en une seule LMI réalisable (solvable avec le toolbox LMI du Matlab).

Exemple :

Considérons les deux LMI multiples suivantes, avec $F : \mathbb{R}^n \rightarrow \mathbb{D}$ est une fonction affine, et pour des matrices A, B et des vecteurs a, b .

$$\begin{cases} F(x) > 0 \\ Ax = a \end{cases} \quad \text{Ou} \quad \begin{cases} F(x) > 0 \\ x = Bu + b, \quad u \in \mathbb{U} \end{cases} \quad (1-7)$$

Plus généralement, ces LMI multiples peuvent être de la forme suivante :

$$\begin{cases} F(x) > 0 \\ x \in \mathcal{M} \end{cases} \quad (1-8)$$

Où \mathcal{M} est un ensemble affine de \mathbb{R}^n , cette LMI multiple peut être transformée en une seule LMI $\tilde{F}(x) > 0$ en éliminant la contrainte affine comme suit :

Tout ensemble affine peut être écrit sous la forme suivante :

$$\mathcal{M} = \{x \in \mathbb{R}^n \mid x = x_0 + m, \quad m \in \mathcal{M}_0\}$$

Avec $x_0 \in \mathbb{R}^n$ et \mathcal{M}_0 un espace linéaire de \mathbb{R}^n et de dimension $\dim(\mathcal{M}_0) = \tilde{n}$ ($\tilde{n} \leq n$) et soit $(e_1, e_2 \dots \dots e_{\tilde{n}})$ sa base, alors en utilisant la décomposition (1-3) la LMI (1-8) s'écrit :

$$\begin{aligned} F(x) > 0 \\ &= F_0 + T(x_0 + \sum_{j=1}^{\tilde{n}} x_j e_j) > 0 \\ &= F_0 + T(x_0) + \sum_{j=1}^{\tilde{n}} x_j T(e_j) > 0 \\ &= \tilde{F}_0 + x_1 \tilde{F}_1 + x_2 \tilde{F}_2 \dots \dots + x_{\tilde{n}} \tilde{F}_{\tilde{n}} > 0 \\ &= \tilde{F}(\tilde{x}) > 0 \end{aligned}$$

Finalement la LMI (1-8) est équivalente à $\tilde{F}(\tilde{x}) > 0$ avec $\tilde{F}_0 = F_0 + T(x_0)$, $\tilde{F}_j = T(e_j)$ et $\tilde{x} = (x_1, x_2 \dots \dots x_{\tilde{n}})^t$ sont les coefficients de $(x - x_0)$ dans la base de \mathcal{M}_0 .

Ces propriétés sont très utiles pour la formulation des questions de stabilité et de stabilisation des systèmes commutés par des LMI à contraintes convexes.

Inégalités matricielles bilinéaires (BMI)

Les BMI représentent une généralisation des LMI, Elles sont des inéquations matricielles qui font intervenir des produits d'inconnus matricielles et scalaires ce qui complique leur résolution. Les BMI ne sont pas des problèmes convexes et peuvent avoir plusieurs solutions locales. Par conséquent, les techniques d'optimisation convexe employées pour résoudre les LMI ne sont pas applicables en général, mais selon la taille du problème en main, les techniques de gridding-up ou de changement de variables pour les variables scalaires sont toujours possibles.

LMI standards :

Parmi les problèmes les plus rencontrés, les trois problèmes suivants généralement associés aux LMI sont pris en charge par le toolbox LMI du Matlab:

- **Problème de réalisabilité ou faisabilité :** il s'agit de trouver l'ensemble S des x tel que $F(x) > 0$.
La façon la plus simple d'attaquer ce problème est de chercher le vecteur x minimisant le scalaire t tel que :

$$-F(x) < tI \quad (1-9)$$

Si la valeur minimale de t est négative, alors le problème est faisable c'est-à-dire l'ensemble S n'est pas vide.

- Problème de valeurs propres (EVP, EigenValue Problems) : il consiste en la minimisation de la plus grande valeur propre d'une matrice symétrique F_1 sous contraintes LMI.

$$\text{minimiser } \lambda \text{ tel que } \begin{cases} \lambda I - F_1(x) > 0 \\ F_2(x) > 0 \end{cases} \quad (1-10)$$

- Problème de valeurs propres généralisées (GEVP, Genalised EigenValue Problems) : il s'agit là de minimiser la plus grande valeur propre généralisée d'une paire de matrices F_1, F_2 sous contraintes LMI.

$$\text{minimiser } \lambda \text{ tel que } \begin{cases} \lambda F_2(x) - F_1(x) > 0 \\ F_2(x) > 0 \\ F_3(x) \end{cases} \quad (1-11)$$

Les deux derniers problèmes peuvent être considérés comme une optimisation de la forme :

$$\min_x f(x) \text{ tel que } \{F(x) > 0 \quad (1-12)$$

Différents types d'algorithmes peuvent être utilisés pour résoudre ces trois problèmes : Simplex, ellipsoïdes ou les méthodes des points intérieurs qui s'avèrent les plus efficaces (voir résolution numériques des LMI).

Complément de Schur

Le complément de Schur est une propriété qui permet de passer d'une inégalité matricielle à une autre, comme c'est déjà vu le résultat est issu directement de l'utilisation de la transformation de congruence :

$$\begin{bmatrix} Q(x) & S(x)^t \\ S(x) & R(x) \end{bmatrix} > 0 \Leftrightarrow \begin{cases} R(x) > 0 \\ Q(x) - S(x)^t R^{-1} S(x) > 0 \end{cases} \Leftrightarrow \begin{cases} Q(x) > 0 \\ R(x) - S(x) Q^{-1} S(x)^t > 0 \end{cases}$$

Avec $Q(x) = Q(x)^t, R(x) = R(x)^t$ et $S(x)$ sont affines en x .

S-Procédure pour des fonctions quadratiques et des inéquations non strictes :

Considérons une collection de $(m+1)$ fonctions quadratiques :

$$F_i(x) = x^t A_i x + b_i^t x + c_i, \quad A_i^t = A_i, \quad i = 0, 1, \dots, m.$$

On veut tester la condition : $F_0(x) > 0$ pour tout x tel que : $F_i(x) \geq 0, \quad i = 1, \dots, m$.

La S-Procédure donne une condition convexe suffisante :

S'il existe $\tau_i \geq 0$ tel que pour tout $x, F_0(x) - \sum_{i=1}^m \tau_i F_i(x) \geq 0$ alors la condition est satisfaite.

Par exemple le cas pour $m=1$: S'il existe un x tel que $F_1(x) > 0$, la réciproque est vraie.

Voici une autre variante n'utilisant que des inéquations strictes et formes quadratiques (fonctions quadratiques sans constantes ni termes de linéarité) :

La condition que : $x^t A_0 x > 0, \text{ pour } x \neq 0 \text{ et } x^t A_i x \geq 0, \quad i = 1, \dots, m$.

Est vraie s'il existe $\tau_i \geq 0$ teque $A_0 - \sum_{i=1}^m \tau_i A_i > 0$.

1.4. Optimisation convexe :

Un problème d'optimisation convexe est généralement de la forme :

$$\text{minimiser } f_0(x) \text{ tel que } \begin{cases} f_i(x) \leq 0, \quad i = 1, \dots, m \\ a_i^t x = b_i, \quad i = 1, \dots, p \end{cases} \quad (1-13)$$

où les fonctions $f_i(x), \quad i = 0, \dots, m$ sont convexes et les fonctions d'égalité $g_i(x) = a_i^t x - b_i, \quad i = 1, \dots, p$ sont affines.

On notera que l'ensemble de réalisabilité de cette optimisation est convexe puisqu'il est égal à l'intersection de m sous-ensembles convexes et de p hyperplans. Le problème restera toujours convexe si seulement la fonction objective $f_0(x)$ est concave et on désire la maximiser. Deux avantages sont à retenir de la formulation de problèmes d'optimisation en optimisation convexe :

- Il n'existe pas de minimum local pour la fonction objective, le résultat obtenu est optimum au sens global.
- De puissants outils numériques existent pour résoudre ce genre de problème en un temps raisonnable.

Heureusement, la majorité des problèmes d'optimisation en théorie des systèmes peuvent être formulés en une optimisation convexe.

Définition: (optimum local et global)

Soit D un sous-ensemble d'un espace normé, un élément x^* est dit optimum (minimum) local d'une fonction $f : D \rightarrow \mathbb{R}$ Si $\forall x \in D, \exists \varepsilon > 0$ tel que :

$$f(x^*) \leq f(x), \text{ pour tout } x: \|x - x^*\| < \varepsilon$$

L'optimum est dit global si $f(x^*) \leq f(x)$, pour tout $x \in D$.

Définition: (optimum de fonction convexe)

Si une fonction $f : D \rightarrow \mathbb{R}$ est convexe, alors toute solution optimale locale de f est une solution optimale globale de f . si f est strictement convexe alors cette solution optimale est unique.

Programmes convexes :

En automatique, on vise essentiellement la stabilité et la stabilisation des systèmes commutés et la formulation LMI de ces problèmes (donc convexe) en se basant sur la théorie de Lyapunov. Comme il est déjà connu, le seul problème avec cette théorie est l'absence de méthodes systématiques pour la recherche des fonctions d'énergies. D'autres part si on échoue de trouver de telle fonction, aucune conclusion ne peut être tirée en ce qui concerne la stabilité des systèmes. C'est dans ce cadre là qu'intervient l'optimisation convexe comme outil mathématique efficace en fournissant une méthode systématique de recherche de fonction de Lyapunov en exploitant les contraintes d'évolution du système.

D'une manière générale, l'optimisation convexe (1-13) appelée aussi programme convexe s'écrit :

$$\text{Min } f(x) \text{ tel que } x \in D \quad (1-14)$$

Où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction convexe et l'ensemble $D \subset \mathbb{R}^n$ est convexe.

Programmes linéaire (LP) :

Un des cas particulier des programmes convexes de type (1-14) est le programme linéaire (LP) dans lequel $f(x)$ est linéaire et l'ensemble D est exprimé par des égalités et inégalités linéaires et possède donc une structure polyédrale.

$$\text{min } c^t x \text{ tel que } Ax \leq b \quad (1-15)$$

Où $x \in \mathbb{R}^{nx}$, $c \in \mathbb{R}^{nx}$, $A \in \mathbb{R}^{m \times nx}$, $b \in \mathbb{R}^m$ et m le nombre de contraintes.

Pour ce type de programme convexe, il existe des méthodes de résolutions numériques efficaces telles que les algorithmes des points intérieurs (voir section ci-après).

Programmes quadratique (QP) :

Un autre type de programme convexe largement utilisé en automatique est le programme quadratique (QP) dans lequel la fonction objective $f(x)$ est quadratique (convexe) toujours avec une structure polyédrale de l'ensemble D .

$$\min x^t Q x + c^t x \quad / \quad A x \leq b \quad (1-16)$$

Où $Q \in \mathbb{R}^{n \times n}$, $Q=Q^t \geq 0$, $x \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ et m le nombre de contraintes.

On notera que les programmes LP sont un cas particulier des QP. De même que les LP, les programmes quadratiques peuvent être résolus en un temps polynomiale mais moins rapidement que les LP.

Programmes semi-défini (SDP) :

Un programme semi-défini est une généralisation des programmes linéaires où les contraintes d'inégalités sont remplacées par des inégalités généralisées de type LMI.

$$\min c^t x \quad \text{tel que} \quad \begin{cases} F_0(x) + \sum_{i=1}^m \tau_i F_i(x) \leq 0, & i = 1, \dots, m \\ A x \leq b \end{cases} \quad (1-17)$$

Probablement c'est le type de programmes le plus exploité en automatique, un exemple typique est la formulation des questions de la stabilité et la stabilisation des systèmes linéaires et non linéaires par le concept de Lyapunov.

Résolution numérique des LMI :

Depuis le début des années 90, les scientifiques ont reconnu que la plupart des problèmes de commande et d'optimisation peuvent se formuler en une optimisation convexe à base de LMI. Toutefois, les algorithmes de l'époque telle que l'algorithme des ellipsoïdes [développé dans les années 70] n'étaient pas efficaces pour la résolution des problèmes convexes. La taille du programme était approximativement une fonction exponentielle du nombre de paramètres, cela a conduit les gens à s'intéresser à l'algorithme rapide de [Karmarkar 1984] initialement développé en 1984 pour les programmes linéaires et son extension aux problèmes d'optimisation à contraintes LMI par [Nesterov 1988]. Depuis, plusieurs variantes et améliorations ont été proposées et l'efficacité de l'algorithme a été reconnue, l'apparition des monographes de [Nesterov & Nemirovski 1994, Boyd & al 1994] font l'apogée de la popularité des LMI et de l'algorithme des points intérieurs. Une propriété importante de la méthode des points intérieurs est leur complexité polynomiale, c'est-à-dire que le nombre d'opérations nécessaires pour résoudre un problème avec une précision donnée est une fonction du nombre d'inconnus et celui des contraintes. Plusieurs outils numériques (solveurs) à base de l'algorithme des points intérieurs sont maintenant disponibles pour résoudre les LMI, la première implémentation software a été gratuitement commercialisée comme un toolbox utilisable avec Matlab (initialement développée pour Scilab en 93) par Gahinet, El Ghaoui en 1995. Un autre solveur populaire est le SeDuMi développé par Sturm en 99 tous deux interfaçables avec le fameux outil YALMIP développé par Loftberg en 2004.

Vue le rôle important des algorithmes des points intérieurs Dans la résolution des problèmes convexes, nous donnons dans la section suivante, le principe de base de la version la plus simple de la méthode.

Méthode des points intérieurs :

D'une manière générale, pour les méthodes des points intérieurs les contraintes sont remplacées par une fonction de pénalité et l'optimisation se poursuit par résolution successive de problèmes sans contraintes. Le principe est le suivant :

Soit F une fonction affine avec $F_i = F_i^t \in \mathbb{R}^{n \times n}$, et la LMI suivante :

$$F(x) = F_0 + x_1 F_1 + \dots + x_m F_m > 0 \quad (1-18)$$

Et soit $D = \{x | F(x) > 0\}$ le domaine d'une fonction convexe $f : D \rightarrow \mathbb{R}$ dont on veut minimiser, ainsi le problème consiste en l'optimisation convexe suivante :

$$x^* = \inf_{x \in D} f(x) \quad \text{Ou équivalamment} \quad \left\{ \begin{array}{l} \min_x f(x) \\ \text{tel que } F(x) > 0 \end{array} \right. \quad (1-19)$$

Pour résoudre ce problème i.e. trouver la/les solutions optimales, il est d'abord nécessaire d'introduire une fonction barrière \emptyset telle que :

- \emptyset est strictement convexe à l'intérieur de D
- Approche $+\infty$ pour toute séquence $\{x_n\}_1^{+\infty}$ dans D qui converge vers les points de bordure (limite) de D .

En utilisant une telle fonction barrière, notre problème d'optimisation à contrainte qui consiste à minimiser f sur tout $x \in D$ est remplacé par l'optimisation sans contrainte qui consiste à minimiser la fonction convexe (au sens stricte dans \mathbb{R}^n) suivante :

$$f_t(x) = tf(x) + \emptyset(x) \quad (1-20)$$

Où $t > 0$ est appelé paramètre de pénalité.

Le principe est de déterminer une application $t \rightarrow x(t)$ qui associe à chaque $t > 0$ un optimum $x_{opt}(t)$ de $f_t(x)$. Puis en variant t , cette optimisation est généralement résolue par la technique itérative de Newton-Raphson pour approximer le minimum x_{opt} de $f_t(x)$. Ainsi, pour une séquence du paramètre de pénalité t_n ($t_n \rightarrow \infty$ quand $t \rightarrow \infty$), on aura x_{opt} ($t_n \rightarrow x_{opt}$ quand $t \rightarrow \infty$) où $x^* = f(x_{opt})$ est solution du problème convexe original (1-19).

Pour l'implémentation, On utilise plutôt souvent pour la minimisation la fonction convexe suivante :

$$g_t(x) = \emptyset_0(t - f(x)) + \emptyset(x) \quad (1-21)$$

Où $t > t_0 = x^*$ et \emptyset_0 est une fonction barrière pour le demi-axe positif des réels. De même, l'objectif serai de déterminer le minimum $x_{opt}(t)$ de $g_t(x)$ pour $t > 0$. C'est-à-dire que à chaque t_i on aura un minimum $x_{opt}(t_i)$ et la trajectoire $x_{opt}(t)$ quand t varie est appelée trajectoire (ou chemin) des centres. Sous certaines conditions $x_{opt}(t)$ est analytique et possède une limite lorsque $t \rightarrow t_0$. Le point $x_{opt} = \lim_{t \rightarrow t_0} (x_{opt}(t))$ est optimal dans le sens où $x^* = f(x_{opt})$ puisque pour $t > t_0$ on a $f(x_{opt}) < t$.

Les méthodes des points intérieurs peuvent être appliquées aux problèmes de réalisabilité et ceux d'optimisation définis précédemment.

Problème de réalisabilité :

Pour le problème de réalisabilité f n'est pas définie et le problème se réduit à :

$$D = \left\{ x \mid F(x) = F_0 + \sum_1^m x_i F_i > 0 \right\}$$

Toujours pour une fonction affine F telle que $F_i = F_i^t \in \mathbb{R}^{n \times n}$, l'objectif est de trouver l'ensemble des $\{x | F(x) > 0\}$. sachant que pour une fonction $F(x)$ affine en x telle que $F_i = F_i^t \in \mathbb{R}^{n \times n}$, la fonction définie par $\log \det F(x)^{-1}$ est une fonction convexe en F , donc la fonction suivante peut être candidate pour servir comme fonction de barrière :

$$\emptyset(x) = \begin{cases} \log \det F(x)^{-1} & \text{pour } x \in D \\ \infty & \text{ailleurs} \end{cases} \quad (1-22)$$

En supposant que l'ensemble D est borné et non vide, ce qui veut dire que $\emptyset(x)$ est strictement convexe dans D et possède un seul minimum (global) noté $\emptyset(x^*)$, le point x^* est appelé centre analytique de l'ensemble de réalisabilité D ou de LMI $F(x) > 0$.

$$\begin{aligned} x^* &= \arg \min_x \emptyset(x) \\ &= \begin{cases} \arg \max_x \log \det F(x) \\ F(x) > 0 \end{cases} \end{aligned} \quad (1-23)$$

Ce minimum est ensuite facilement calculable par la technique itérative de Newton-Raphson qui converge quadratiquement vers le centre analytique x^* .

Problème d'optimisation :

En générale, le problème d'optimisation consiste en une minimisation sous contrainte de la forme (1-19) :

$$\begin{cases} \min_x f(x) \\ F(x) > 0 \end{cases} \quad \text{Ou équivalamment} \quad \begin{cases} \min_x f(x) \\ x \in D \end{cases}$$

Plus précisément, il s'agit de trouver: $x^* = \inf_{x \in S} f(x)$.

Ce problème peut être facilement transformé en un problème de réalisabilité comme suit :

$$\tilde{F}_t(x) = \begin{pmatrix} t - f(x) & 0 \\ 0 & F(x) \end{pmatrix} > 0 \quad (1-24)$$

Où comme mentionné précédemment, $t > t_0 = x^*$ est le paramètre de pénalité. En utilisant toujours la fonction barrière (1-22) pour la LMI (1-24), on aura l'optimisation sans contrainte qui consiste à minimiser la fonction strictement convexe suivante :

$$g_t(x) = \log \det \tilde{F}_t(x)^{-1} = \underbrace{\log \frac{1}{(t-f(x))}}_{\emptyset_0(t-f(x))} + \underbrace{\log \det F(x)^{-1}}_{\emptyset(x)} \quad (1-25)$$

La séquence du minimum global $x_{opt}(t_i)$ de $g_t(x)$ converge vers $x^* = \inf_{x \in S} f(x)$ quand $t \rightarrow t_0$.

Note :

Il existe des versions d'algorithme d'implémentation plus récentes et plus rapides de la méthode des points intérieurs que nous n'avons pas pu consulter. D'autre part, il y a aussi des solveurs spécialisées qui sont sophistiquées donc plus efficaces et peuvent prendre en charges certains types de BMI ou des LMI à contraintes d'égalités, ces solveurs ne sont pas commercialisés et ce n'est pas facile de se procurer d'eux. À ce niveau de graduation, Nous pouvons se contenter des résultats d'optimisation pouvant être obtenus par le LMI du Matlab.