



DÉPARTEMENT STPI

3ÈME ANNÉE MIC

Introduction à l'Optimisation Numérique

Aude Rondepierre & Sébastien Tordeux

Année 2009-2010

Table des matières

Introduction	5
1 Formulation et analyse d'un problème d'optimisation	7
1.1 Notions de minimum, maximum, infimum, supremum	7
1.2 Description d'un problème d'optimisation	8
1.2.1 Formulation	8
1.2.2 Ensemble des contraintes	11
1.3 Convexité et optimisation	12
1.3.1 Éléments d'analyse convexe	12
1.3.2 Résultats d'unicité en optimisation convexe	14
2 Optimisation numérique sans contraintes	15
2.1 Condition suffisante d'existence d'un point minimum	15
2.2 Conditions d'optimalité	16
2.3 Généralités sur les algorithmes de descente	18
2.3.1 Notion de direction de descente	18
2.3.2 Algorithme de descente modèle	20
2.3.3 Convergence et vitesse de convergence	22
2.4 Premiers algorithmes de descente	23
2.4.1 Algorithmes de gradient à pas fixe/pas optimal	23
2.4.2 Méthode de Newton locale	25
2.4.3 Méthode de Gauss-Newton	29
3 Introduction à l'optimisation sous contraintes	31
3.1 Condition suffisante d'existence d'un point minimum	31
3.2 Conditions d'optimalité	33
3.3 Algorithme du gradient projeté	34
A Compléments	39
A.1 Rappels de calcul différentiel	39
A.1.1 Différentiabilité et gradient	39
A.1.2 Dérivées d'ordre supérieur	39
A.1.3 Formules de Taylor	40
A.2 Quelques démonstrations	42

A.2.1	Méthode de Gauss-Newton pour la résolution des problèmes de moindres carrés	42
-------	---	----

Introduction

Tout problème d'optimisation comporte une étape essentielle : la modélisation mathématique. Elle consiste en trois étapes :

1. Identification des variables de décisions (souvent désignées par un vecteur $x \in \mathbb{R}^n$ dans ce cours) : ce sont les paramètres sur lesquels l'utilisateur peut agir pour faire évoluer le système considéré.
2. Définition d'une fonction coût (appelée fonction objectif) permettant d'évaluer l'état du système (ex : rendement, performance, ...).
3. Description des contraintes imposées aux variables de décision.

Le problème d'optimisation consiste alors à déterminer les variables de décision conduisant aux meilleures conditions de fonctionnement du système (ce qui revient à minimiser ou maximiser la fonction coût), tout en respectant les contraintes d'utilisation définies à l'étape 3.

Dans ce cours, nous nous intéressons aux méthodes numériques pour l'optimisation continue, différentiable et non linéaire.

Le premier chapitre traite des notions mathématiques fondamentales à maîtriser avant de s'intéresser à la résolution à proprement parler de tout problème d'optimisation : la description mathématique d'un problème d'optimisation, la notion de solution locale et quelques éléments d'analyse convexe.

Dans les chapitres suivants, nous entrons dans le vif du sujet en nous intéressant à la résolution de problèmes d'optimisation sans contrainte (cf chapitre 2), puis avec contraintes (cf chapitre 3). Pour chacun de ses problèmes nous tenterons de répondre aux questions suivantes :

1. Existe-t-il une solution (même locale) du problème considéré ? si oui, a-t-on unicité ?
2. Comment la caractériser ? (cf conditions d'optimalité).
3. Comment la calculer ? Quel type d'algorithme choisir ?

Chapitre 1

Formulation et analyse d'un problème d'optimisation

1.1 Notions de minimum, maximum, infimum, supremum

On distinguera les notions de minimum et de maximum des notions d'infimum et de supremum. Ces notions sont des prérequis pour les démonstrations des résultats d'existence et d'unicité d'extrema d'une fonction donnée.

Définition 1.1 (Minorant/Majorant) Soit E un sous-ensemble de \mathbb{R} .

$m \in \mathbb{R} \cup \{-\infty, +\infty\}$ est un minorant de E ssi

$$\forall x \in E \quad m \leq x.$$

$M \in \mathbb{R} \cup \{-\infty, +\infty\}$ est un majorant de E ssi

$$\forall x \in E \quad M \geq x.$$

Définition 1.2 (Infimum/Supremum) Soit $E \subset \mathbb{R}$.

L'infimum $\inf(E) \in \mathbb{R} \cup \{-\infty, +\infty\}$ de E est le plus grand des minorants.

Le supremum $\sup(E) \in \mathbb{R} \cup \{-\infty, +\infty\}$ de E est le plus petit des majorants. On note aussi

$$\inf_{x \in E}(x) \quad \text{et} \quad \sup_{x \in E}(x).$$

Définition-Proposition 1.1 (Suite minimisante/Suite maximisante) Soit $E \subset \mathbb{R}$, $E \neq \emptyset$.

Il existe $(x_n)_n$ une suite d'éléments de E telle que

$$\lim_{n \rightarrow +\infty} x_n = \inf(E)$$

et une suite $(y_n)_n$ d'éléments de E telle que

$$\lim_{n \rightarrow +\infty} y_n = \sup(E).$$

La suite $(x_n)_n$ est une suite minimisante et $(y_n)_n$ une suite maximisante de E .

Preuve. Comme $E \neq \emptyset$, nécessairement : $\inf(E) \in \mathbb{R} \cup \{-\infty\}$ et $\sup(E) \in \mathbb{R} \cup \{+\infty\}$.

1er cas : $\inf(E) \in \mathbb{R}$. Comme $\inf(E)$ est le plus grand des minorants de E , alors quel que soit $n \in \mathbb{N}$, $\inf(E) + 1/n$ n'est pas un minorant de E . Il existe donc un élément $x_n \in E$ tel que

$$x_n \leq \inf(E) + 1/n.$$

Cette suite (x_n) d'éléments de E vérifie $\inf(E) \leq x_n \leq \inf(E) + 1/n$ et admet donc $\inf(E)$ comme limite.

2nd cas : $\inf(E) = -\infty$. E admet seulement $-\infty$ comme minorant. Par conséquent pour tout $n \in \mathbb{N}$, il existe $x_n \in E$ tel que

$$x_n \leq -n.$$

La suite x_n ainsi construite converge vers $-\infty$. La construction de la suite (y_n) est laissée en exercice. \square

Définition 1.3 (Minimum, maximum) Soit $E \subset \mathbb{R}$. Un infimum est un minimum si $\inf(E) \in E$. Un supremum est un maximum si $\sup(E) \in E$.

Dans ce cas, on note

$$\min(E) = \inf(E) \quad \text{et} \quad \max(E) = \sup(E).$$

1.2 Description d'un problème d'optimisation

1.2.1 Formulation

On considère le problème d'optimisation suivant :

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{sous la contrainte : } x \in X. \quad (1.1)$$

où X est un sous-ensemble de \mathbb{R}^n . La fonction $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$, appelée *fonction objectif*, est supposée au moins différentiable. L'ensemble X est appelé *ensemble des contraintes*.

Le problème (1.1) est dit *réalisable* si $X \neq \emptyset$.

Un cas fréquent en optimisation est celui où X est défini par des égalités et des inégalités :

$$X = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\}$$

où $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ et $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$ sont supposées continues. L'écriture " $h(x) = 0$ " représente p contraintes d'égalité :

$$h_i(x) = 0, \quad i = 1, \dots, p,$$

et de même " $g(x) \leq 0$ " représente q contraintes d'inégalités :

$$g_i(x) \leq 0, \quad i = 1, \dots, q.$$

Dans ce cas, on dit qu'il s'agit d'un problème d'optimisation à n variables de décisions, p contraintes d'égalités et q contraintes d'inégalités.

Résoudre le problème (1.1) revient à chercher des points de minimum local (ou global, c'est encore mieux !) au sens de la définition suivante :

Définition 1.4 (Minimum local/Minimum global) Soit $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ une fonctionnelle.

- $x \in \mathbb{R}^n$ est un point de minimum local de f sur X si

$$x \in X \quad \text{et} \quad \exists r > 0 \mid \forall y \in X \cap B(x, r)^1, \quad f(x) \leq f(y). \quad (1.2)$$

On dit alors que $f(x)$ est un minimum local de f sur X .

- $x \in \mathbb{R}^n$ est un point de minimum global de f sur X ssi

$$x \in X \quad \text{et} \quad \forall y \in X, \quad f(x) \leq f(y). \quad (1.3)$$

On dit alors que $f(x)$ est un minimum global de f sur X .

Les notions de maximum local et global sont définies de façon tout à fait similaire. En fait, on peut facilement démontrer que les problèmes (avec ou sans contraintes) :

$$\min_x f(x) \quad \text{et} \quad \max_x -f(x)$$

sont équivalents dans le sens où ils ont même ensemble de solutions et :

$$\min_x f(x) = -\max_x -f(x) \quad \text{ou encore} \quad \max_x f(x) = -\min_x -f(x).$$

Ainsi la recherche d'un maximum pouvant se ramener à la recherche d'un minimum, nous porterons une attention plus particulière à la recherche du minimum.

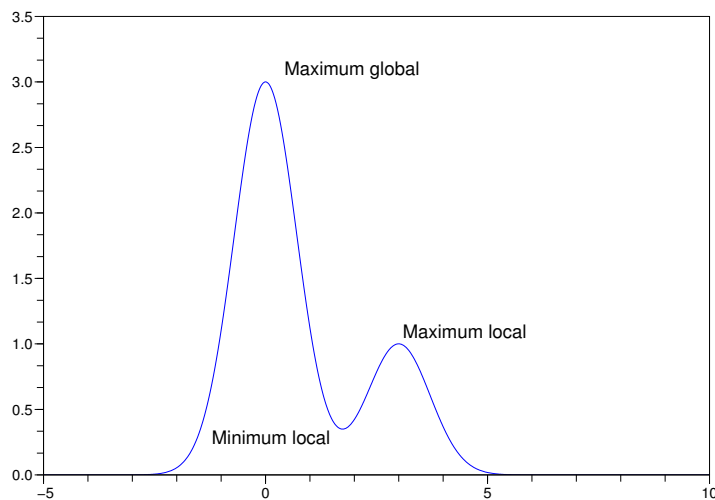


FIGURE 1.1 – $f : x \mapsto 3 * e^{-x^2} + e^{-(x-3)^2}$: exemples de minima et maxima locaux/globaux

1. On note $B(x, r) = \{y \in \mathbb{R}^n \mid \|y - x\| < r\}$ la boule ouverte de centre x et de rayon r .

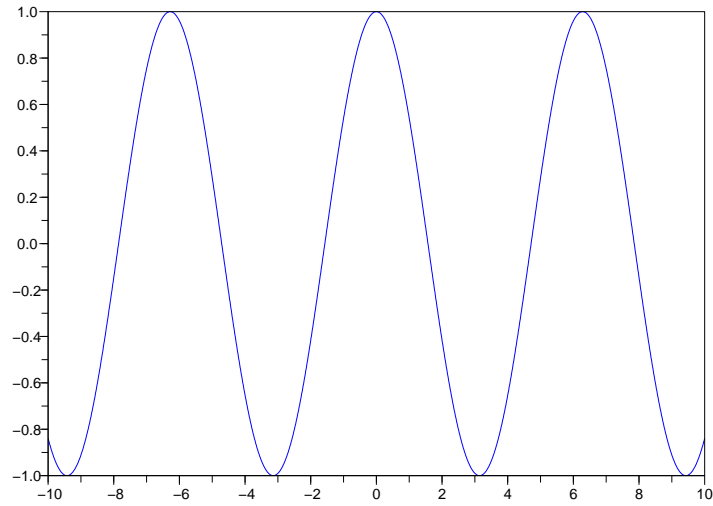


FIGURE 1.2 – Fonction $f : x \mapsto \cos(x)$: infinité de minima et maxima globaux.

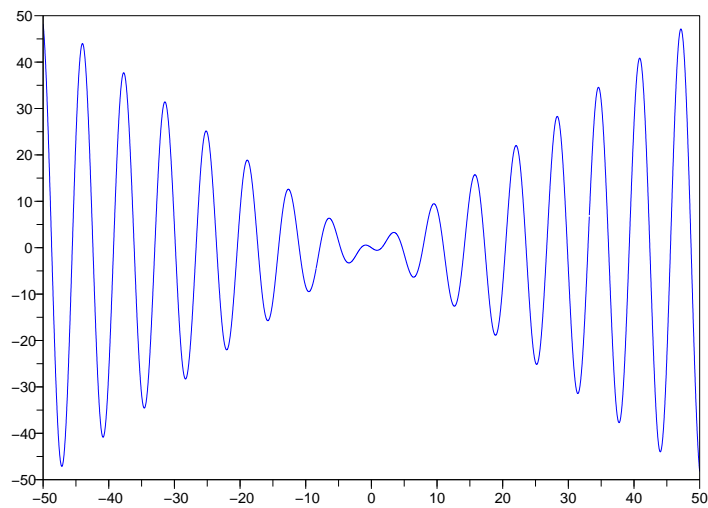


FIGURE 1.3 – Fonction $f : x \mapsto x \cos(x)$: infinité de minima et maxima locaux mais aucun minimum ou maximum global.

1.2.2 Ensemble des contraintes

Commençons cette section par quelques notions élémentaires de topologie dans \mathbb{R}^n .

Définition 1.5 (Ensemble ouvert) Soit $O \subset \mathbb{R}^n$. L'ensemble O est ouvert si pour tout $x \in O$, il existe une boule ouverte $B(x, r) = \{y \in \mathbb{R}^n \mid \|y - x\| < r\}$ de centre x et de rayon r telle que :

$$B(x, r) \subset O$$

Exemple 1.2.1 1. Les ensembles \emptyset et \mathbb{R}^n sont ouverts.

2. Les boules ouvertes sont des ouverts.

3. Les intervalles de la forme $]a, b[$, $-\infty \leq a < b \leq +\infty$, sont des ouverts de \mathbb{R} .

Définition 1.6 (Ensemble fermé) Soit $F \subset \mathbb{R}^n$. L'ensemble F est un fermé si son complémentaire est un ouvert.

Exemple 1.2.2

1. Les ensembles \emptyset et \mathbb{R}^n sont fermés, leurs complémentaires respectifs étant \mathbb{R}^n et \emptyset .

2. Les intervalles de la forme $[a, b]$, $-\infty < a < b < +\infty$, sont des fermés de \mathbb{R} .

Proposition 1.1 Soit $F \subset \mathbb{R}^n$. F est un fermé si et seulement si toute suite convergente d'éléments de F a sa limite dans F .

Preuve. (\Rightarrow) Supposons que F est fermé et notons O son complémentaire. Soit $(x_n)_{n \in \mathbb{N}}$ une suite d'éléments de F qui converge vers $x \in \mathbb{R}^n$. Supposons par l'absurde que $x \in O$.

Comme O est ouvert, il existe $r > 0$ tel que : $B(x, r) \subset O$. Or $(x_n)_{n \in \mathbb{N}}$ converge vers x d'où :

$$\exists N \in \mathbb{N}, \forall n \geq N, \|x_n - x\| < r,$$

Ceci implique $x_n \in B(x, r) \subset O$ et donc $x_n \in O$ à partir d'un certain rang, ce qui est impossible car $x_n \in F$ quel que soit $n \in \mathbb{N}$ par définition.

(\Leftarrow) Supposons que toute suite convergente de F admet une limite dans F . Montrons que O , le complémentaire de F , est ouvert.

Par l'absurde, supposons que O ne soit pas ouvert, i.e. : il existe $x \in O$ tel que : $\forall r > 0$, $B(x, r) \not\subset O$. Autrement dit :

$$\forall r > 0 \quad B(x, r) \cap F \neq \emptyset.$$

Pour $r = \frac{1}{n+1}$, nous construisons une suite $(x_n)_{n \in \mathbb{N}}$ d'éléments de F telle que

$$x_n \in B(x, \frac{1}{n+1}).$$

Ceci peut être traduit en $\|x_n - x\| \leq \frac{1}{n+1}$. Il suit que x_n converge vers x . Comme F est fermé, $x \in F$, ce qui est impossible car $x \in O$. \square

Revenons maintenant à la caractérisation de l'ensemble des contraintes dans le cas de contraintes d'égalité et/ou d'inégalité.

Proposition 1.2 Soient $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$ et $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ deux fonctions continues.

- $X = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\}$ est un ensemble fermé de \mathbb{R}^n .
- $X = \{x \in \mathbb{R}^n : g(x) < 0\}$ est un ouvert de \mathbb{R}^n .

1.3 Convexité et optimisation

Les problèmes dont les données sont convexes, constituent une classe importante en optimisation, car fréquemment rencontrés dans les applications et à la base de nombreuses méthodes développées pour des problèmes plus généraux.

1.3.1 Éléments d'analyse convexe

Définition 1.7 (Ensemble convexe) Soit $C \subset \mathbb{R}^n$. L'ensemble C est convexe ssi

$$\forall (x, y) \in C^2, \quad \forall \lambda \in]0, 1[, \quad \lambda x + (1 - \lambda)y \in C,$$

c'est-à-dire, si x et y sont deux éléments de C alors le segment qui relie x à y est inclus dans C .

Exemple. \mathbb{R}^n est convexe.

Définition 1.8 (Fonction convexe/strictement convexe) Soit $C \subset \mathbb{R}^n$ convexe et $f : C \rightarrow \mathbb{R}$.

- f est convexe ssi

$$\forall (x, y) \in C^2, \quad \forall \lambda \in]0, 1[, \quad f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

- f est strictement convexe ssi

$$\forall (x, y) \in C^2, \quad x \neq y, \quad \forall \lambda \in]0, 1[, \quad f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y).$$

Théorème 1.1 Soit $C \subset \mathbb{R}^n$ convexe et $f : C \rightarrow \mathbb{R}$ différentiable. La fonction f est convexe ssi :

$$\forall (x, y) \in C^2, \quad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \quad (1.4)$$

ou de façon équivalente, ssi :

$$\forall (x, y) \in C^2, \quad \langle \nabla f(y) - \nabla f(x), y - x \rangle \geq 0. \quad (1.5)$$

Preuve. Soit $(x, y) \in C^2$. Par convexité de f , on a donc pour tout $t \in]0, 1[$:

$$f((1 - t)x + ty) \leq (1 - t)f(x) + tf(y) = f(x) + t(f(y) - f(x)),$$

soit : $\frac{f(x + t(y - x)) - f(x)}{t} \leq f(y) - f(x)$. En passant à la limite pour $t \rightarrow 0^+$, il suit (1.4).

Réciproquement, on applique (1.4) à $tx + (1 - t)y$ et x , puis à $tx + (1 - t)y$ et y , d'où :

$$\begin{aligned} f(x) &\geq f(tx + (1 - t)y) + (1 - t) \langle \nabla f(tx + (1 - t)y), y - x \rangle \\ f(y) &\geq f(tx + (1 - t)y) - t \langle \nabla f(tx + (1 - t)y), y - x \rangle \end{aligned}$$

En combinant ces deux inégalités, on obtient : $tf(x) + (1 - t)f(y) \geq f(tx + (1 - t)y)$, et donc la convexité de f .

En échangeant les rôles de x et y dans (1.4), puis en sommant les deux inégalités obtenues, on démontre sans problème que (1.4) implique (1.5). Pour montrer la réciproque, on introduit :

$$\varphi : t \in [0, 1] \mapsto tf(x) + (1 - t)f(y) - f(tx + (1 - t)y),$$

et on montre que φ est positive sur $[0, 1]$. Démonstration laissée en exercice (on pourra tracer le tableau de variation de φ sur $[0, 1]$). \square

Remarque 1.1 Si ∇f est strictement monotone i.e. si les inégalités (1.4) et (1.5) sont strictes pour $x \neq y$, alors f est strictement convexe.

Si de plus, la fonctionnelle f est deux fois différentiable, on a alors une caractérisation d'ordre deux de la convexité via la Hessienne.

Théorème 1.2 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonctionnelle de classe \mathcal{C}^2 .

Si la hessienne $H[f](x)$ de f est une matrice symétrique définie positive pour tout $x \in \mathbb{R}^n$, alors f est strictement convexe.

Si $H[f](x)$ est une matrice symétrique semidéfinie positive pour tout $x \in \mathbb{R}^n$, alors f est convexe.

Pour montrer ce résultat nous avons besoin du lemme technique suivant.

Lemme 1.1 Soit $\phi \in \mathcal{C}^2([0, 1])$.

- Si $\phi^{(2)}(x) > 0$, alors $\phi(\lambda) < (1 - \lambda)\phi(0) + \lambda\phi(1)$ pour $\lambda \in]0, 1[$.
- Si $\phi^{(2)}(x) \geq 0$, alors $\phi(\lambda) \leq (1 - \lambda)\phi(0) + \lambda\phi(1)$.

Preuve. Comme $\phi^{(2)}(x) > 0$ (resp. \geq) pour $0 \leq x \leq 1$, ϕ' est strictement croissante (resp. croissante) et par conséquent

$$\begin{aligned}\phi(\lambda) - \phi(0) &= \int_0^\lambda \phi'(t) dt < \lambda\phi'(\lambda) \quad (\text{resp. } \leq), \\ \phi(1) - \phi(\lambda) &= \int_\lambda^1 \phi'(t) dt > (1 - \lambda)\phi'(\lambda) \quad (\text{resp. } \geq).\end{aligned}$$

En groupant ces deux inégalités, nous tirons

$$\frac{\phi(\lambda) - \phi(0)}{\lambda} < \left(\phi'(\lambda) < \right) \frac{\phi(1) - \phi(\lambda)}{1 - \lambda} \quad (\text{resp. } \leq);$$

c'est-à-dire : $\phi(\lambda) < (1 - \lambda)\phi(0) + \lambda\phi(1)$ (resp. \leq). □

Preuve du théorème 1.2. Supposons $H[f](z)$ symétrique définie positive. Soient x et y deux éléments distincts de \mathbb{R}^n . Introduisons la fonction $\phi : [0, 1] \rightarrow \mathbb{R}$ de classe \mathcal{C}^2 définie par

$$\phi(\lambda) = f(\lambda x + (1 - \lambda)y).$$

Utilisons le développement à l'ordre deux de ϕ pour calculer la dérivée seconde de ϕ

$$\phi(\lambda + h) = f((\lambda + h)x + (1 - \lambda - h)y) = f(\lambda x + (1 - \lambda)y + h(x - y)).$$

Avec $t = \lambda x + (1 - \lambda)y$, nous avons

$$\begin{aligned}\phi(\lambda + h) &= f(t) + [\nabla f(t)]^\top h(x - y) + \frac{1}{2}h(x - y)^\top H[f](t)h(x - y) \\ &\quad + h^2 \|x - y\|^2 \varepsilon(t, h(x - y)),\end{aligned}$$

et, par conséquent

$$\phi(\lambda + h) = \phi(\lambda) + h \left([\nabla f(t)]^\top (x - y) \right) + \frac{h^2}{2} \left((x - y)^\top H[f](t)(x - y) \right) + h^2 \varepsilon(h)$$

Comme ϕ est de classe C^2 , la dérivée première et la dérivée seconde de ϕ sont donc données par

$$\begin{aligned}\phi'(\lambda) &= [\nabla f(\lambda x + (1 - \lambda)y)]^\top (x - y), \\ \phi^{(2)}(\lambda) &= (x - y)^\top \underbrace{H[f](\lambda x + (1 - \lambda)y)}_{\in \mathbb{R}^{n \times n}} (x - y).\end{aligned}$$

Donc la fonction $\phi^{(2)}(\lambda)$ est strictement positive car $H[f](\lambda x + (1 - \lambda)y)$ est symétrique définie positive et $x \neq y$. Le lemme 1.1 s'écrit :

$$f(\lambda x + (1 - \lambda)y) < (1 - \lambda)f(y) + \lambda f(x).$$

puisque : $\phi(0) = f(y)$, $\phi(1) = f(x)$ et $\phi(\lambda) = f(\lambda x + (1 - \lambda)y)$.

Ceci prouve que f est strictement convexe. Si pour tout x , $H[f](x)$ est positive, une preuve similaire nous permet de montrer que f est convexe. \square

1.3.2 Résultats d'unicité en optimisation convexe

Théorème 1.3 (Condition suffisante d'optimalité globale) Soient $C \subset \mathbb{R}^n$ un convexe et $f : C \rightarrow \mathbb{R}$ une fonctionnelle. Soit x^* un point de minimum local de f .

- i. Si f est convexe, alors x^* est un point de minimum global de f .
- ii. Si f est strictement convexe, alors x^* est l'unique point de minimum global de f .

Preuve. i. Par l'absurde : soit $x^* \in C$ un point qui réalise un minimum local de f sur C , i.e. :

$$\exists r > 0 \mid \forall y \in C \text{ avec } \|y - x^*\| < r, \quad f(y) \geq f(x^*). \quad (1.6)$$

tel que x^* ne réalise pas un minimum global, i.e. qu'il existe un point x^+ de C tel que :

$$f(x^+) < f(x^*). \quad (1.7)$$

On introduit le point : $y_{r/2} = \alpha x^+ + (1 - \alpha)x^*$, avec : $\alpha = \frac{r}{2\|x^+ - x^*\|} \in]0, 1[$.

D'après les hypothèses (1.6) et (1.7), on a : $x^+ \notin B(x^*, r)$, ce qui implique : $r < \|x^+ - x^*\|$, et : $\alpha \in]0, 1[$. Par convexité de C , le point $y_{r/2}$ appartient donc au segment $[x^+, x^*]$ lui-même contenu dans C . De plus :

$$\begin{aligned}f(y_{r/2}) &\leq \alpha f(x^+) + (1 - \alpha)f(x^*) && \text{par convexité de } f \\ &< f(x^*) && \text{d'après (1.6)}.\end{aligned}$$

Ceci contredit (1.6) car : $\|y_{r/2} - x^*\| = \frac{r}{2}$. Le point $y_{r/2}$ appartient donc également à la boule ouverte $B(x^*, r)$ i.e. : $f(y_{r/2}) \geq f(x^*)$.

ii. Raisonnons à nouveau par l'absurde. Soient x_1 et x_2 deux éléments de C réalisant le minimum de f . Par convexité de C , $\frac{x_1 + x_2}{2} \in C$, et comme f est strictement convexe, il suit

$$\begin{aligned}f\left(\frac{x_1 + x_2}{2}\right) &< \frac{1}{2}f(x_1) + \frac{1}{2}f(x_2) \\ &< \frac{1}{2}\inf_{y \in C} f(y) + \frac{1}{2}\inf_{y \in C} f(y) = \inf_{y \in C} f(y),\end{aligned}$$

Ce qui est impossible. \square

Chapitre 2

Optimisation numérique sans contraintes

Nous nous intéressons dans ce chapitre à la conception de méthodes numériques pour la recherche des points $x \in \mathbb{R}^n$ qui réalisent le minimum d'une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$(P) \quad \min_{x \in \mathbb{R}^n} f(x),$$

où f est supposée au moins différentiable. On parle d'optimisation sans contrainte.

2.1 Condition suffisante d'existence d'un point minimum

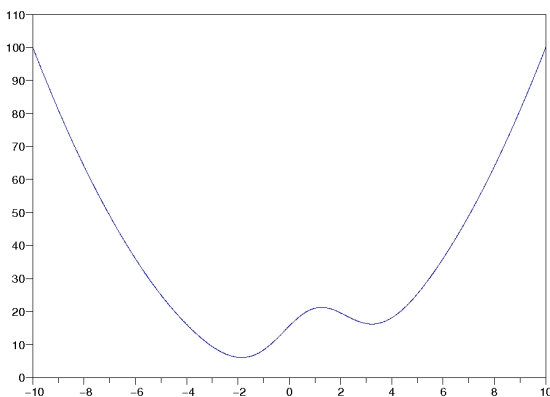
Avant d'étudier la solution (ou les solutions) de (P) , il faut s'assurer de leur existence.

Définition 2.1 Une application $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est dite *infinie à l'infini* (ou *coercive*) ssi

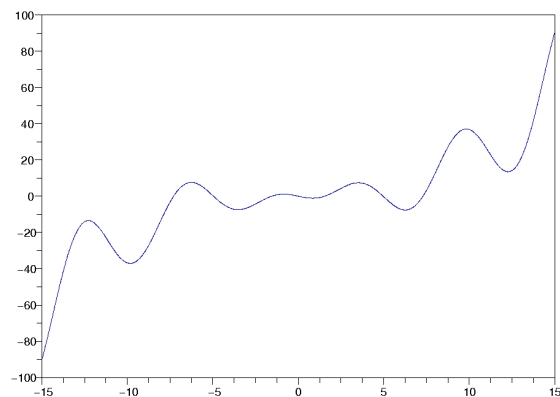
$$\forall A \in \mathbb{R}, \quad \exists R > 0 \mid \forall x \in X, \quad [\|x\| \geq R \implies f(x) \geq A] \quad (2.1)$$

On note : $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$.

Exemples



Exemple de fonction coercive



Exemple de fonction non coercive

1. $f(x) = \|x\|_2$ est coercive.

2. $f(x) = x_1^2 - x_2^2$ n'est pas coercive : en effet, la suite de terme général $x_n = (0, n)$, $n \in \mathbb{N}$, est telle que : $\lim_{n \rightarrow +\infty} \|x_n\| = \lim_{n \rightarrow +\infty} n = +\infty$ mais : $\lim_{n \rightarrow +\infty} f(x_n) = \lim_{n \rightarrow +\infty} -n^2 = -\infty$.

Pour montrer que f est infinie à l'infini on utilise souvent la proposition suivante :

Proposition 2.1 Soit $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ une application et $g : \mathbb{R} \rightarrow \mathbb{R}$ vérifiant

$$f(x) \geq g(\|x\|) \quad \text{avec} \quad \lim_{t \rightarrow +\infty} g(t) = +\infty.$$

Alors, f est infinie à l'infini.

Preuve. Comme g tend vers $+\infty$ en $+\infty$

$$\forall A \in \mathbb{R}, \quad \exists R > 0 \mid \forall t \in \mathbb{R} \quad t \geq R \implies g(t) \geq A.$$

Avec $t = \|x\|$ et comme $g(x) \geq f(\|x\|)$, nous obtenons (2.1). □

Théorème 2.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continue et infinie à l'infini. Alors il existe un point $x \in \mathbb{R}^n$ qui réalise le minimum de f sur \mathbb{R}^n . Autrement dit, il existe $x \in \mathbb{R}^n$ tel que

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

Preuve. Soit $d = \inf_{x \in \mathbb{R}^n} f(x) < +\infty$. Soit $(x_n)_{n \in \mathbb{N}}$ une suite minimisante c'est-à-dire telle que :

$$\lim_{n \rightarrow +\infty} f(x_n) = d < +\infty. \quad (2.2)$$

Montrons que la suite $(x_n)_{n \in \mathbb{N}}$ est bornée. Par l'absurde, on suppose qu'elle ne l'est pas c'est-à-dire qu'il existe une sous-suite notée $(x_{\varphi(n)})_n$ de $(x_n)_n$ telle que : $\lim_{n \rightarrow +\infty} \|x_{\varphi(n)}\| = +\infty$. Par coercivité de f , on a alors : $\lim_{n \rightarrow +\infty} f(x_{\varphi(n)}) = +\infty$, ce qui contredit (2.2).

La suite $(x_n)_{n \in \mathbb{N}}$ est donc bornée : il existe alors une suite extraite notée $(x_{\psi(n)})_n$ de $(x_n)_n$, qui converge vers $\bar{x} \in \mathbb{R}^n$. En utilisant maintenant la continuité de f , on a alors :

$$f(\bar{x}) = \lim_{n \rightarrow +\infty} f(x_{\psi(n)}) = d.$$

On en déduit alors deux choses : $d > -\infty$ et \bar{x} solution du problème (P). □

2.2 Conditions d'optimalité

Théorème 2.2 (Conditions nécessaires d'optimalité locale) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une application différentiable. Si $x^* \in \mathbb{R}^n$ réalise un minimum local (resp. maximum local) de f , alors :

$$\nabla f(x^*) = 0 \quad (\text{CN d'optimalité du } 1^{\text{er}} \text{ ordre})$$

Si, de plus, f est deux fois différentiable dans un voisinage ouvert de x^* , alors :

$$\begin{aligned} & H[f](x^*) \text{ semidéfinie positive} && (\text{CN d'optimalité du } 2^{\text{nd}} \text{ ordre}) \\ & (\text{resp. } H[f](x^*) \text{ semidéfinie négative}) \end{aligned}$$

Preuve. Soit $h \in \mathbb{R}^n$, $h \neq 0$. Pour s assez petit, on définit $\varphi : s \in \mathbb{R} \mapsto f(x^* + sh)$. φ admet donc un minimum local en $s = 0$, d'où : $\varphi'(0) = \nabla f(x^*)^\top h = 0$. Ceci étant vrai pour tout h , on en déduit : $\nabla f(x^*) = 0$.

Supposons maintenant f deux fois différentiable. On écrit le développement de Taylor d'ordre 2 de la fonction φ . Comme $\nabla f(x^*) = 0$, on obtient :

$$f(x^* + sh) - f(x^*) = \frac{s^2}{2} h^\top H[f](x^*) h + o(s^2).$$

soit : $\frac{s^2}{2} h^\top H[f](x^*) h + o(s^2) \geq 0$ puisque x^* est un point de minimum local de f . Après division par s^2 , on fait tendre s vers 0 et on obtient : $h^\top H[f](x^*) h \geq 0$. \square

La condition du premier ordre joue un rôle central en optimisation numérique : elle permet de sélectionner un certain nombre de points candidats à être des extrema locaux, appelés **points critiques** ou **points stationnaires**. Parmi eux, figurent des minima locaux, des maxima locaux et d'autres qui ne sont ni l'un, ni l'autre, appelés *points selle*.

Définition 2.2 (Points critiques) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une application différentiable. Tout point $x \in \mathbb{R}^n$ vérifiant :

$$\nabla f(x) = 0,$$

est appelé *point critique* (ou *point stationnaire*) de f .

Mais attention ! Les conditions du théorème 2.2 ne sont que nécessaires : tout point où le gradient est nul n'est pas nécessairement un extremum. En effet, la fonctionnelle : $f : x \in \mathbb{R} \mapsto x^3$ admet en $x = 0$ un point de dérivée nulle, qui n'est pas un extremum même local de f .

Théorème 2.3 (Condition Suffisante d'optimalité locale) Soit O un ouvert de \mathbb{R}^n . Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une application supposée de classe C^2 sur O . Si $\bar{x} \in O$ vérifie :

$$\nabla f(\bar{x}) = 0 \quad \text{et} \quad H[f](\bar{x}) \text{ symétrique, définie positive (resp. définie négative)}$$

Alors \bar{x} est un point de minimum local (resp. maximum local) de f .

Remarque 2.1

1. Comme f est de classe C^2 , par le théorème de Schwarz, il n'est pas nécessaire de vérifier que la matrice hessienne $H[f]$ est symétrique.
2. D'un point de vue géométrique, la condition du second ordre : $H[f](\bar{x})$ définie positive, revient à dire que f est localement convexe en x^* , i.e. convexe dans un voisinage ouvert de x^* . En pratique, elle est difficile à vérifier systématiquement car elle nécessite de calculer les dérivées secondes et d'étudier les valeurs propres de la matrice hessienne.

Si de plus la fonctionnelle à optimiser est convexe ou strictement convexe, en appliquant le théorème 1.3 au convexe $C = \mathbb{R}^n$, on obtient :

Théorème 2.4 (Condition Suffisante d'optimalité globale) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une application différentiable et \bar{x} un point critique de f .

- i. Si f est convexe, alors \bar{x} est un point de minimum global de f .
- ii. Si f est strictement convexe, alors \bar{x} est l'unique point de minimum global de f .

2.3 Généralités sur les algorithmes de descente

Partant d'un point x_0 arbitrairement choisi, un algorithme de descente va chercher à générer une suite d'itérés $(x_k)_{k \in \mathbb{N}}$ telle que :

$$\forall k \in \mathbb{N}, \quad f(x_{k+1}) \leq f(x_k).$$

Commençons par définir plus précisément la notion de descente.

2.3.1 Notion de direction de descente

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

Définition 2.3 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une application continue. Soit $x \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$.

La dérivée directionnelle de f en x dans la direction d est définie par :

$$df(x; d) := \lim_{t \rightarrow 0^+} \frac{f(x + td) - f(x)}{t},$$

si cette limite existe.

Proposition 2.2 Si f est différentiable en un point $x \in \mathbb{R}^n$, alors pour tout $d \neq 0$, f admet une dérivée dans la direction d en x et :

$$df(x; d) = Df(x)(d) = \nabla f(x)^\top d.$$

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

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

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

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

Définition 2.4 (Direction de descente) Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $x \in \mathbb{R}^n$. Le vecteur $d \in \mathbb{R}^n$ est une direction de descente pour f à partir du point x si $t \mapsto f(x + td)$ est décroissante en $t = 0$, c'est-à-dire s'il existe $\eta > 0$ tel que :

$$\forall t \in]0, \eta], \quad f(x + td) < f(x). \quad (2.3)$$

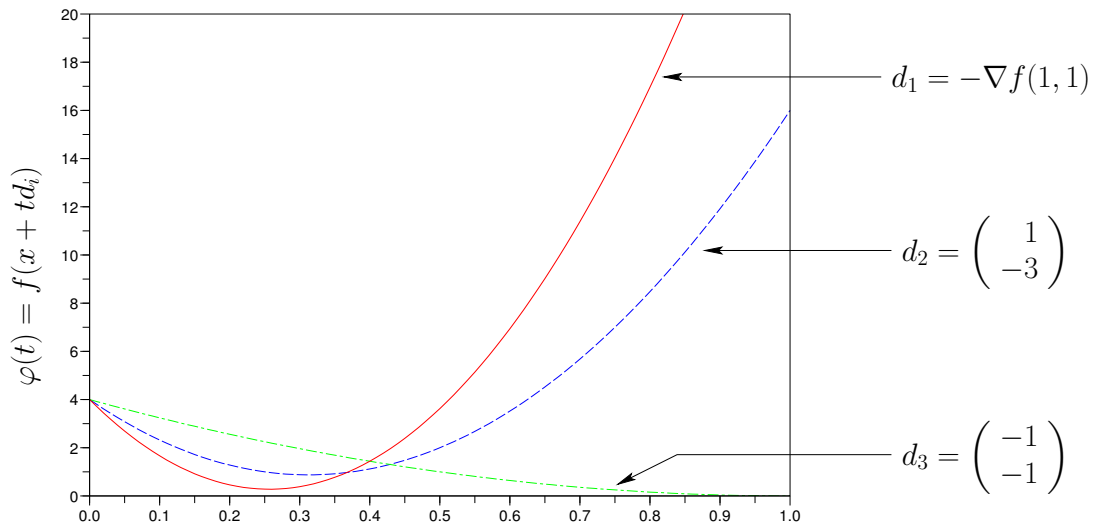


FIGURE 2.1 – Allure de la fonction $f : x \mapsto \frac{1}{2}x_1^2 + 2x_2^2$ au point $x = (1, 1)^\top$ dans plusieurs directions.

Proposition 2.3 Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable et $x \in \mathbb{R}^n$ tel que : $\nabla f(x) \neq 0$. Le vecteur $d \in \mathbb{R}^n$ est une direction de descente pour f à partir du point x ssi la dérivée directionnelle de f en x dans la direction d vérifie :

$$df(x; d) = \nabla f(x)^\top d < 0. \tag{2.4}$$

De plus pour tout $\beta < 1$, il existe $\bar{\eta} > 0$ tel que :

$$\forall t \in]0, \bar{\eta}], f(x + td) < f(x) + t\beta \nabla f(x)^\top d. \tag{2.5}$$

Autrement dit, la relation (2.5) nous dit que la décroissance de la fonction objectif f en faisant un pas de taille t dans la direction d , est d’au moins le pas fois une fraction β de la pente.

Preuve de la proposition 2.3. Soit d une direction de descente de f en x , soit : $\nabla f(x)^\top d < 0$. On écrit le développement de Taylor-Young de $t \mapsto f(x + td)$: pour t assez petit,

$$f(x + td) = f(x) + t\nabla f(x)^\top d + t\epsilon(t), \text{ avec } : \epsilon(t) \xrightarrow[t \rightarrow 0]{} 0.$$

Sachant que $\varepsilon = -\nabla f(x)^\top d > 0$, il existe $\eta > 0$ tel que si $|t| \leq \eta$, alors $|\epsilon(t)| < \varepsilon$. D’où :

$$\forall t \in]0, \eta], f(x + td) - f(x) = t [\nabla f(x)^\top d + \epsilon(t)] < t [\nabla f(x)^\top d + \varepsilon] = 0.$$

On démontre de la même façon l'inégalité (2.5), en choisissant : $\varepsilon = (\beta - 1)\nabla f(x)^\top d$ dans la définition de Landau, et $\beta < 1$. \square

Parmi toutes les directions de descente existant en un point x donné, une des plus remarquables est celle où la pente est la plus forte, c'est-à-dire selon le gradient. Pour le démontrer, il suffit de comparer les dérivées directionnelles :

Théorème 2.5 (Direction de plus forte descente) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Soit $x \in \mathbb{R}^n$. Alors pour toute direction d de norme constante égale à $\|d\| = \|\nabla f(x)\|$, on a :

$$(-\nabla f(x))^\top \nabla f(x) \leq d^\top \nabla f(x), \quad (2.6)$$

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

Preuve. Soit $d \in \mathbb{R}^n$ une direction quelconque de norme : $\|d\| = \|\nabla f(x)\|$. On a alors :

$$\begin{aligned} (-d)^\top \nabla f(x) &\leq \| -d \| \|\nabla f(x)\| && \text{d'après l'inégalité de Cauchy-Schwarz} \\ &\leq \|\nabla f(x)\|^2 = \nabla f(x)^\top \nabla f(x) && \text{puisque } \|d\| = \|\nabla f(x)\|. \end{aligned}$$

On en déduit donc (2.6). \square

Sans surprise, si l'opposé du gradient correspond à la plus forte descente, le gradient correspond, lui, à la plus forte pente :

Corollaire 2.1 Le vecteur $\nabla f(x)$ est appelée direction de plus forte pente de f au point $x \in \mathbb{R}^n$.

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

Exercice 2.3.1 Retrouver les conditions nécessaires d'optimalité du théorème 2.2 en utilisant la notion de direction de descente.

2.3.2 Algorithme de descente modèle

Partant d'un point x_0 arbitrairement choisi, un algorithme de descente va chercher à générer une suite d'itérés $(x_k)_{k \in \mathbb{N}}$ telle que :

$$\forall k \in \mathbb{N}, \quad f(x_{k+1}) \leq f(x_k)$$

D'après la caractérisation de la descente (cf proposition 2.3), il s'agit donc à chaque itération k , de trouver un point x_{k+1} dans une direction d vérifiant : $\nabla f(x_k)^\top d < 0$.

Le schéma général d'un algorithme de descente est le suivant :

ALGORITHME DE DESCENTE MODÈLE.

Données: $f : \mathbb{R}^n \rightarrow \mathbb{R}$ supposée au moins différentiable, x_0 point initial arbitrairement choisi.

Sortie: une approximation de la solution du problème : $\min_{x \in \mathbb{R}^n} f(x)$.

1. $k := 0$
 2. Tant que “test de convergence” non satisfait,
 - (a) Trouver une direction de descente d_k telle que : $\nabla f(x_k)^\top d_k < 0$.
 - (b) *Recherche linéaire* : Choisir un pas $s_k > 0$ à faire dans cette direction et tel que :

$$f(x_k + s_k d_k) < f(x_k).$$
 - (c) Mise à jour : $x_{k+1} = x_k + s_k d_k$; $k := k + 1$;
 3. Retourner x_k .
-

Oracle/Boîte noire. Pour obtenir le prochain itéré, l’algorithme aura besoin d’informations sur la fonction objectif f : la valeur numérique de f en un point donné x , et souvent également le gradient $\nabla f(x)$. Ces informations sont fournies en “boîte noire”, i.e. par un sous-programme indépendant de l’algorithme d’optimisation choisi : routine de calcul du gradient par différences finies lorsque celui-ci n’est pas calculable explicitement, ou simulateur renvoyant les valeurs numériques $f(x)$ et $\nabla f(x)$ sans formule mathématique explicite par exemple.

Test de convergence/Test d’arrêt. Soit x^* un point de minimum local du critère f à optimiser. Supposons que l’on choisisse comme test d’arrêt dans l’algorithme de descente modèle, le critère idéal : “ $x_k = x^*$ ”. Dans un monde idéal (i.e. en supposant tous les calculs exacts et la capacité de calcul illimitée), soit l’algorithme s’arrête après un nombre fini d’itérations, soit il construit (théoriquement) une suite infinie $x_1, x_2, \dots, x_k, \dots$ de points de \mathbb{R}^n qui converge vers x^* .

En pratique, un test d’arrêt devra être choisi pour garantir que l’algorithme s’arrête toujours après un nombre fini d’itérations et que le dernier point calculé soit suffisamment proche de x^* .

Soit $\varepsilon > 0$ la précision demandée. Plusieurs critères sont à notre disposition : tout d’abord (et c’est le plus naturel), un critère d’optimalité basé sur les conditions nécessaires d’optimalité du premier ordre présentées dans la section 2.2 : on teste si

$$\|\nabla f(x_k)\| < \varepsilon, \quad (2.7)$$

auquel cas l’algorithme s’arrête et fournit l’itéré courant x_k comme solution.

En pratique, le test d’optimalité n’est pas toujours satisfait et on devra faire appel à d’autres critères (fondés sur l’expérience du numérique) :

- Stagnation de la solution : $\|x_{k+1} - x_k\| < \varepsilon \|x_k\|$.
- Stagnation de la valeur courante : $\|f(x_{k+1}) - f(x_k)\| < \varepsilon |f(x_k)|$.
- Nombre d’itérations dépassant un seuil fixé à l’avance : $k < \text{IterMax}$.

et généralement une combinaison de ces critères :

Critère d’arrêt = Test d’optimalité satisfait
 OU (Stagnation de la valeur courante & Stagnation de la solution)
 OU Nombre d’itérations maximum autorisé dépassé.

En pratique, on préférera travailler avec les erreurs relatives plutôt qu’avec les erreurs absolues, trop dépendantes de l’échelle.

2.3.3 Convergence et vitesse de convergence

Étudier la convergence d'un algorithme, c'est étudier la convergence de la suite des itérés générés par l'algorithme. Un algorithme de descente selon le modèle précédent, est dit *convergent* si la suite de ses itérés $(x_k)_{k \in \mathbb{N}}$ converge vers un point limite x^* , solution du problème :

$$\min_{x \in \mathbb{R}^n} f(x).$$

De plus, la convergence est dite *locale* si elle n'a lieu que pour des points initiaux x_0 dans un voisinage de x^* . Sinon elle est dite *globale*.

En pratique, le but d'un algorithme d'optimisation ne sera que de trouver un point critique (i.e. un point vérifiant la condition d'optimalité du premier ordre : $\nabla f(x^*) = 0$). On introduit alors la notion de convergence globale d'un algorithme d'optimisation :

Définition 2.5 Soit un algorithme itératif qui génère une suite $(x_k)_{k \in \mathbb{N}}$ dans \mathbb{R}^n afin de résoudre le problème :

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

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une application de classe C^1 . L'algorithme est dit *globalement convergent* si quel que soit le point initial $x_0 \in \mathbb{R}^n$,

$$\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

Cette propriété garantit que le critère d'arrêt " $\|\nabla f(x_k)\| \leq \varepsilon$ " sera satisfait à partir d'un certain rang quelle que soit la précision $\varepsilon > 0$ demandée.

Il est bien entendu très important de garantir la convergence d'un algorithme sous certaines hypothèses, mais la vitesse de convergence et la complexité sont également des facteurs à prendre en compte lors de la conception ou de l'utilisation d'un algorithme ; en effet, on a tout intérêt à ce que la méthode choisie soit à la fois rapide, précise et stable. Pour cela, on introduit les notions de vitesse (ou taux) de convergence qui mesurent l'évolution de l'erreur commise $\|x_k - x^*\|$.

Définition 2.6 Soit $(x_k)_{k \in \mathbb{N}}$ une suite d'itérés générés par un algorithme convergent donné. On note x^* la limite de la suite $(x_k)_{k \in \mathbb{N}}$ et on suppose : $\forall k \in \mathbb{N}, x_k \neq x^*$ (sinon l'algorithme convergerait en un nombre fini d'itérations). La convergence de l'algorithme est dite :

– *linéaire* si l'erreur $e_k = \|x_k - x^*\|$ décroît linéairement i.e. s'il existe $\tau \in]0, 1[$ tel que :

$$\lim_{k \rightarrow +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \tau.$$

– *superlinéaire* si

$$\lim_{k \rightarrow +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

– *d'ordre p* s'il existe $\tau \geq 0$ tel que :

$$\lim_{k \rightarrow +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} = \tau.$$

En particulier, si $p = 2$, la convergence est dite *quadratique* (grosso modo à partir d'un certain rang, le nombre de chiffres significatifs exacts double à chaque itération).

Bien entendu, on a intérêt à ce que la convergence d'un algorithme soit la plus élevée possible afin de converger vers la solution en un minimum d'itérations pour une précision donnée.

Exemple 2.3.1 La fonction $f : x \mapsto x^3 - 6x + 1$ admet un minimum local sur \mathbb{R} en $x^* = \sqrt{2}$. Partant d'une approximation grossière $x_0 = 2$ de x^* , comparons plusieurs algorithmes de calcul approché de x^* avec 5 chiffres significatifs exacts :

- Soit l'algorithme $x_{k+1} = x_k - \alpha(x_k^2 - 2)$. Vérifier que pour $0 < \alpha < \frac{1}{\sqrt{2}}$, cet algorithme converge linéairement avec un taux $\tau = |2\alpha\sqrt{2} - 1|$.

α	$\frac{2}{3}$	0.5	$\frac{1}{3}$	$\frac{1}{2\sqrt{2}}$
$\tau = 2\alpha\sqrt{2} - 1 $	0.885	0.414	0.057	0
Nb d'itérations	105	15	6	4
Nb chiffres sign. exacts	5	5	7	10

Si $\alpha = \frac{1}{2\sqrt{2}}$, la convergence est dite superlinéaire et c'est la meilleure convergence possible de l'algorithme en question.

- Soit l'algorithme : $x_{k+1} = \frac{1}{2}(x_k + \frac{2}{x_k})$ dont la convergence est quadratique. Alors 4 itérations suffisent pour calculer une valeur approchée de x^* avec 5 chiffres significatifs exacts ; en réalité, on a même 11 chiffres significatifs exacts dès la quatrième itération.

2.4 Premiers algorithmes de descente

Un algorithme de descente est déterminé par les stratégies de choix des directions de descente successives, puis par le pas qui sera effectué dans la direction choisie. Concentrons nous dans cette partie sur le choix de la direction de descente : l'idée est de remplacer f par un modèle local plus simple, dont la minimisation nous donnera une direction de descente de f .

2.4.1 Algorithmes de gradient à pas fixe/pas optimal

Soit $x_k \in \mathbb{R}^n$ l'itéré courant. Étant donné la valeur $f(x_k)$ et le gradient $\nabla f(x_k)$ (notre "oracle"), on remplace f au voisinage de x_k par son développement de Taylor de premier ordre :

$$f(x_k + d) \sim f(x_k) + \nabla f(x_k)^\top d.$$

On voudrait que la dérivée directionnelle $\nabla f(x_k)^\top d$ soit la plus petite possible dans un voisinage de $d = 0$. On cherche donc à résoudre :

$$\min_{d \in \mathbb{R}^n} \nabla f(x_k)^\top d \quad \text{s.c.} \quad \|d\| = 1,$$

dont la solution nous ait donnée par le théorème 2.5, comme la direction de plus forte descente normalisée :

$$d_k = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}.$$

Le choix de la direction de plus forte descente définit une famille d'algorithmes appelés algorithmes de descente de gradient dont le schéma est le suivant :

ALGORITHME DE DESCENTE DE GRADIENT.

Données: f , x_0 première approximation de la solution cherchée, $\varepsilon > 0$ précision demandée.

Sortie: une approximation x^* de la solution de : $\nabla f(x) = 0$.

1. $k := 0$;
 2. Tant que critère d'arrêt non satisfait,
 - (a) *Direction de descente* : $d_k = -\nabla f(x_k) / \|\nabla f(x_k)\|$.
 - (b) *Recherche linéaire* : trouver un pas s_k tel que : $f(x_k + s_k d_k) < f(x_k)$.
 - (c) $x_{k+1} = x_k - s_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$; $k := k + 1$;
 3. Retourner x_k .
-

Il reste maintenant à définir une stratégie de recherche linéaire pour le calcul du pas. Nous étudions ici en première approche une méthode à pas optimal, puis une à pas fixe.

Méthode de plus profonde descente (“Steepest descent”)

Une idée naturelle consiste à suivre la direction de plus forte descente et à faire un pas qui rende la fonction à minimiser la plus petite possible dans cette direction. Cette méthode est appelée méthode de gradient à pas optimal ou encore **méthode de plus profonde descente**. L'étape 2(a) de l'algorithme de descente de gradient est alors remplacée par :

RECHERCHE LINÉAIRE EXACTE.

2. (a) Calculer un pas optimal s_k solution de : $\min_{s>0} f(x_k + s d_k)$.
-

La méthode de plus profonde descente est une sorte d'idéalisation : d'une part, nous ne savons pas en pratique calculer de façon exacte un point minimum s_k de l'objectif dans une direction donnée et le problème n'est en général pas trivial. D'autre part, la résolution du problème de minimisation unidimensionnel de l'étape 2 (a), même de façon approchée, coûte cher en temps de calcul. Pour ces raisons, on peut lui préférer parfois l'algorithme de gradient à pas constant.

Algorithme de gradient à pas fixe

L'idée est très simple : on impose une fois pour toutes la taille du pas effectué selon la direction de descente calculée à chaque itération. Les itérations 2 (a) et (b) de l'algorithme de descente de gradient sont alors remplacées par :

$$x_{k+1} = x_k - s \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}.$$

La question est alors : comment choisir un pas qui garantisse la convergence de l'algorithme ?

Quelques observations numériques. On souhaite minimiser $f : (x, y) \in \mathbb{R}^2 \mapsto \frac{1}{2}x^2 + \frac{7}{2}y^2$, en utilisant les algorithmes de descente de gradient à pas fixe et pas optimal.

Commençons par analyser le problème de minimisation : d'une part, la fonction f est deux fois différentiable sur \mathbb{R}^2 et strictement convexe. D'autre part, le point $(0, 0)$ vérifie les conditions suffisantes d'optimalité du théorème 2.4. Donc $(0, 0)$ est l'unique point de minimum global de f .

Soit $X_k = (x_k, y_k) \in \mathbb{R}^2$ l'itéré courant tel que : $\nabla f(x_k, y_k) \neq 0$. Calculons par la méthode de plus profonde descente, l'itéré suivant :

– Direction de plus forte descente : $d_k = -\nabla f(X_k) = \begin{pmatrix} -x_k \\ -7y_k \end{pmatrix}$.

– Calcul du pas optimal s_k solution, si elle existe, du problème à une dimension :

$$\min_{s>0} f(X_k + sd_k) = \min_{s>0} \frac{1}{2}x_k^2(1-s)^2 + \frac{7}{2}y_k^2(1-7s)^2.$$

La solution se calcule de façon immédiate : $s_k = (x_k^2 + 7^2y_k^2)/(x_k^2 + 7^3y_k^2)$.

A chaque itération, la méthode génère donc le point : $x_{k+1} = x_k + \frac{x_k^2 + 7^2y_k^2}{x_k^2 + 7^3y_k^2} \begin{pmatrix} -x_k \\ -7y_k \end{pmatrix}$.

Appliquons maintenant ces deux méthodes à partir du point $x_0 = (7, 1.5)$. Leurs comportements sont illustrés par la figure 2.2 et les itérations sont décrites dans les tableaux 2.1 et 2.2

Cet exemple met en évidence la lenteur de la méthode de plus profonde descente, caractérisée par le comportement en zigzag des itérés. Essayons de comprendre d'où vient ce phénomène.

A l'itération $k + 1$, l'algorithme de plus profonde descente minimise $\varphi : s \in \mathbb{R} \mapsto f(x_k - s\nabla f(x_k))$. L'objectif f étant supposé différentiable, la fonction φ est dérivable sur \mathbb{R} de dérivée :

$$\varphi'(s) = -\nabla f(x_k)^\top \nabla f(x_k - s\nabla f(x_k)).$$

Soit s_k le pas optimal calculé ; nécessairement s_k vérifie : $\varphi'(s_k) = 0$, soit :

$$\nabla f(x_k)^\top \nabla f(x_k - s_k \nabla f(x_k)) = 0.$$

Le point $x_{k+1} = x_k - s_k \nabla f(x_k)$ vérifie donc : $\nabla f(x_k)^\top \nabla f(x_{k+1}) = 0$.

Deux directions de descente successives calculées par l'algorithme de plus profonde descente sont **orthogonales** ce que traduisent les zigzags des itérés, observés sur la figure 2.2.

Enfin, les données du tableau 2.2 illustrent l'importance du choix du pas dans l'algorithme de pas fixe : un pas "bien choisi" donne des résultats comparables à ceux obtenus par la plus profonde descente, un pas plus petit atténue les zigzag des itérés mais augmente significativement le nombre d'itérations et enfin, un pas trop grand fait diverger la méthode.

2.4.2 Méthode de Newton locale

Pour construire les méthodes de gradient, nous avons remplacé f par son approximation linéaire au voisinage de l'itéré courant. Nous avons vu que ces méthodes ne sont pas très performantes, en partie parce qu'elles ne tiennent pas compte de la courbure qui est une information de second ordre.

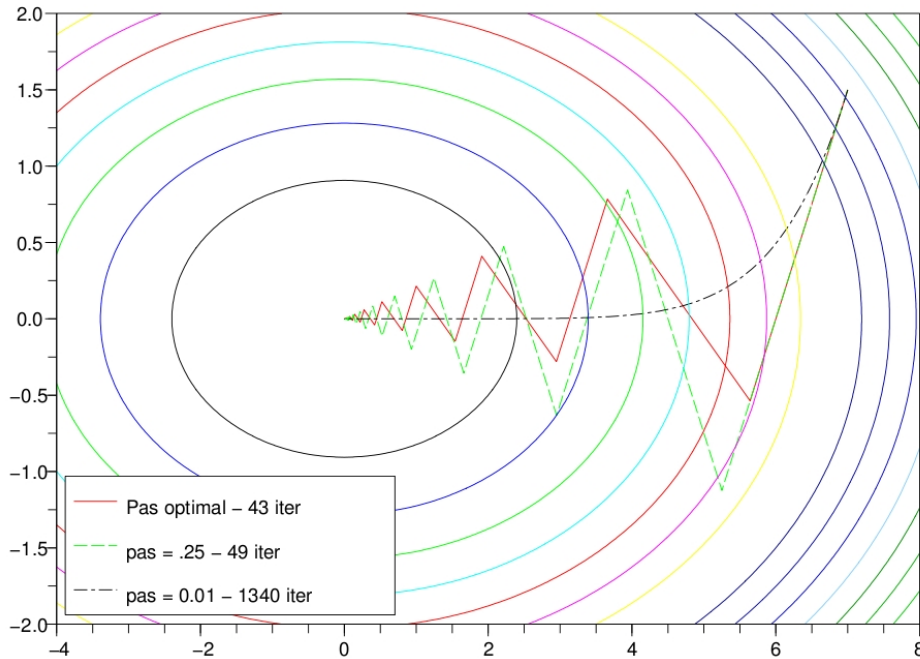


FIGURE 2.2 – Itérations des algos de gradient pas fixe et optimal, générées à partir du point (7, 1.5).

Principe

Supposons maintenant que f est de classe C^2 et remplaçons f au voisinage de l'itéré courant x_k par son développement de Taylor de second ordre :

$$f(y) \sim q(y) = f(x_k) + \nabla f(x_k)^\top (y - x_k) + \frac{1}{2} (y - x_k)^\top H[f](x_k) (y - x_k),$$

où la valeur $f(x_k)$, le gradient $\nabla f(x_k)$ et la matrice hessienne $H[f](x_k)$ sont donnés par notre oracle (boite noire).

On choisit alors comme point x_{k+1} le minimum de la quadratique q lorsqu'il existe et est unique, ce qui n'est le cas que si $H[f](x_k)$ est définie positive. Or le minimum de q est réalisé par x_{k+1} solution de : $\nabla q(x_{k+1}) = 0$, soit :

$$\nabla f(x_k) + H[f](x_k)(x_{k+1} - x_k) = 0,$$

ou encore, en supposant que $H[f](x_k)$ est définie positive :

$$x_{k+1} = x_k - H[f](x_k)^{-1} \nabla f(x_k). \quad (2.8)$$

On reconnaît dans la formule (2.8) les itérations de la méthode de Newton vue en cours d'analyse numérique, appliquée ici à la résolution de l'équation : $\nabla f(x) = 0$. La méthode ne doit cependant jamais être appliquée en utilisant une inversion de la matrice Hessienne (qui peut être de très grande taille et mal conditionnée) mais plutôt en utilisant :

k	$f(x_k, y_k)$	$\ \nabla f(x_k, y_k)\ _2$	s_k	x_k	y_k
0	32.375	10.547512	—	7	1.5
1	16.925373	7.9786973	0.1940299	5.641791	-0.5373134
2	8.8484403	6.5973298	0.3513514	3.6595401	0.7841872
3	4.6258889	3.5448339	0.1940299	2.9494801	-0.2809029
4	2.4183752	3.4490276	0.3513514	1.9131763	0.4099663
5	1.2643059	1.8532089	0.1940299	1.541963	-0.1468536
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
40	$1.751e-10$	2.9343653×10^{-5}	0.3513514	1.63×10^{-5}	0.35×10^{-5}
41	$9.155e-11$	1.5725775×10^{-5}	0.1940299	1.31×10^{-5}	-0.12×10^{-5}
42	$4.786e-11$	1.536522×10^{-5}	0.3513514	0.85×10^{-5}	0.18×10^{-5}
43	$2.502e-11$	0.8292768×10^{-5}	0.1940299	0.69×10^{-5}	0.07×10^{-5}
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
76	$1.268e-20$	0.2523886×10^{-9}	0.3513514	0.14×10^{-9}	0.03×10^{-9}
77	$6.630e-21$	0.1303840×10^{-9}	0.1940299	0.11×10^{-9}	-0.01×10^{-9}
78	$3.466e-21$	0.1303840×10^{-9}	0.3513514	0.72×10^{-10}	0.16×10^{-10}
79	$1.812e-21$	$0.6989278 \times 10^{-10}$	0.1940299	0.58×10^{-10}	-0.05×10^{-10}

TABLE 2.1 – Itérations de la méthode de plus profonde descente. Le critère d’optimalité est satisfait en 43 itérations pour une précision $\varepsilon = 10^{-5}$ et en 79 itérations si $\varepsilon = 10^{-10}$.

pas	0.325	0.25	0.125	0.05	0.01
Nb d’itérations	DV	49	101	263	1340

TABLE 2.2 – Nombres d’itérations de l’algorithme de gradient à pas fixe pour approcher l’unique argument minimum de f à 10^{-5} près, en fonction du pas choisi - Point initial : $x_0 = (7, 1.5)$.

$x_{k+1} = x_k + d_k$
où d_k est l’unique solution du système linéaire :
$H[f](x_k)d_k = -\nabla f(x_k).$
d_k est appelée direction de Newton .

Cette méthode est bien définie si à chaque itération, la matrice hessienne $H[f](x_k)$ est définie positive : ceci est vrai en particulier au voisinage de la solution x^* cherchée si on suppose que $H[f](x^*)$ est définie positive (par continuité de $H[f]$).

Algorithme

MÉTHODE DE NEWTON LOCALE.

Données: $f : \mathbb{R}^n \rightarrow \mathbb{R}$ de classe C^2 , x_0 première approximation de la solution cherchée, $\varepsilon > 0$ précision demandée.

Sortie: une approximation x^* de la solution.

1. $k := 0$;
 2. Tant que $\|\nabla f(x_k)\| > \varepsilon$,
 - (a) Calculer d_k solution du système : $H[f](x_k)d_k = -\nabla f(x_k)$;
 - (b) $x_{k+1} = x_k + d_k$;
 - (c) $k := k + 1$;
 3. Retourner x_k ;
-

Remarque 2.2 Deux observations importantes :

1. La méthode de Newton est un algorithme de descente à pas fixe égal à 1.
2. Si la fonctionnelle f est quadratique, strictement convexe, alors l'algorithme converge en une itération.

Exercice 2.4.1 Démontrer les deux assertions de la remarque 2.2.

Convergence de la méthode de Newton locale

L'algorithme hérite des propriétés de l'algorithme de Newton vu en cours d'analyse numérique pour la résolution des équations non linéaires :

Proposition 2.4 Soit f de classe C^3 et x^* un point de minimum local de f . On suppose en outre que la matrice Hessienne $H[f](x^*)$ est définie positive.

1. Il existe un voisinage \mathcal{V}^* de x^* tel que si $x_0 \in \mathcal{V}^*$, alors la suite des itérés $(x_k)_{k \in \mathbb{N}}$ générés à partir de x_0 par la méthode de Newton locale, converge vers x^* .
2. La convergence est au moins quadratique.

La méthode peut diverger si le point initial n'est pas suffisamment proche d'un point de minimum local, et elle n'est pas définie si les matrices $H[f](x_k)$ ne sont pas définies positives. Utilisée dans le cadre de l'optimisation, la méthode de Newton locale présente un autre inconvénient : la solution identifiée à la fin de l'algorithme n'est pas forcément un point de minimum local, mais uniquement un point critique de f .

2.4.3 Méthode de Gauss-Newton

Si maintenant F désigne une application de \mathbb{R}^n dans \mathbb{R}^m , avec par exemple $m > n$, le système d'équations $F(x) = 0$ n'a généralement pas de solutions. Le problème de moindres carrés associé à F consiste à rechercher x^* tel que

$$r(x^*) = \min \left\{ r(x) = \frac{1}{2} \sum_{i=1}^m F_i(x)^2 = \frac{1}{2} \|F(x)\|_2^2, x \in \mathbb{R}^n \right\}. \quad (2.9)$$

De tels problèmes se rencontrent fréquemment dans le cadre de l'identification de paramètres. Les variables x_i sont les n paramètres d'un modèle physique non linéaire. On effectue $m > n$ mesures, et on cherche les x_i qui permettent d'ajuster au mieux ce modèle aux mesures.

La solution de (2.9) est caractérisée par $\nabla r(x^*) = 0$. Pour appliquer la méthode de Newton, on doit résoudre des systèmes de la forme

$$H_r(x) d = -\nabla r(x) \iff \left(J_F(x)^T J_F(x) + \sum_{i=1}^m F_i(x) H_{F_i}(x) \right) d = -J_F(x)^T F(x), \quad (2.10)$$

où $J_F(x) = \nabla F(x)^T = [\nabla F_1(x) \dots \nabla F_m(x)]^T$ désigne la matrice Jacobienne de F en x . Les calculs permettant d'obtenir la relation (2.10) sont détaillés en annexe.

La matrice Hessienne $H_r(x)$ de $r(x)$ a une expression assez compliquée. Cependant le terme $\sum_{i=1}^m F_i(x) H_{F_i}(x)$ est tel que lorsque le résidu $\|F(x)\|$ devient petit, c'est-à-dire lorsque l'on se rapproche de la solution, il devient lui même négligeable.

La méthode de Gauss-Newton consiste à remplacer (2.10) par :

$$J_F(x)^T J_F(x) d = -J_F(x)^T F(x). \quad (2.11)$$

ALGORITHME DE GAUSS-NEWTON.

Données: F fonction différentiable, x_0 point initial, $\varepsilon > 0$ précision demandée.

Sortie: une approximation de la solution du problème de moindres carrés :

$$\min_{x \in \mathbb{R}^n} r(x) = \frac{1}{2} F(x)^T F(x).$$

1. $k := 0$;

2. Tant que ,

(a) *Calcul d'une direction de recherche :* calculer d_{k+1} solution de :

$$J_F(x)^T J_F(x) d = -J_F(x)^T F(x).$$

(b) $x_{k+1} = x_k + d_{k+1}$;

(c) $k := k + 1$;

3. Retourner s_k .

Application aux moindres carrés linéaires. Dans le cas où la fonction F est linéaire, i.e. :

$$F(x) = Ax - b, \quad \text{avec} \quad A \in M_{n,p}(\mathbb{R}),$$

on obtient : $J_F(x) = A$, et l'équation de Gauss-Newton (2.11) devient : $A^\top A d_{k+1} = -A^\top (Ax_k - b)$. Comme $d_{k+1} = x_{k+1} - x_k$, on obtient :

$$A^\top A x_{k+1} = A^\top b$$

et ceci quel que soit x_k . On reconnaît ici le système d'équations normales du problème de moindres carrés linéaire :

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2, \quad (2.12)$$

On rappelle que d'après le théorème 2.4 du cours d'analyse numérique, x^* est solution du problème (2.12) si et seulement si x^* vérifie les équations normales : $A^\top Ax = A^\top b$. De plus si A est de rang plein, x^* est l'unique solution de (2.12).

Ceci signifie donc que la méthode de Gauss-Newton identifie la solution en une seule itération lorsque la fonction F est linéaire.

Exercice 2.4.2 Soit $F : \mathbb{R}^3 \rightarrow \mathbb{R}^n$ une application définie par :

$$F_i(x_0, x_1, x_2) = c_i - x_0 - x_1 e^{-x_2 t_i}, \quad i = 1, \dots, n.$$

1. Calculer ∇F_i et $H[F_i]$.
2. En déduire : J_r et H_r où $r = \frac{1}{2} F(x)^\top F(x)$.

Chapitre 3

Introduction à l'optimisation sous contraintes

Ce chapitre est une courte introduction à l'optimisation sous contrainte. On s'intéresse à la résolution de problèmes d'optimisation de la forme :

$$\min f(x) \quad \text{s.c.} \quad x \in X, \quad (3.1)$$

où X est un sous-ensemble non vide de \mathbb{R}^n .

TERMINOLOGIE :

- L'ensemble X est appelé *ensemble ou domaine des contraintes*.
- Tout point $x \in \mathbb{R}^n$ vérifiant : $x \in X$, est appelé *point admissible* du problème (3.1).

Chercher une solution du problème avec contraintes (3.1) revient à chercher un point de minimum local de f dans l'ensemble des points admissibles.

3.1 Condition suffisante d'existence d'un point minimum

La première question est celle de l'existence du point de minimum global de la fonction f sur C . Il existe principalement deux théorèmes qui permettent de répondre à cette question. Le premier affirme l'existence d'un point de minimum lorsque l'ensemble des contraintes est fermé borné. Le second est son équivalent pour les ensembles de contraintes fermés mais non bornés.

Théorème 3.1 (Théorème de Weierstrass) *Soit X un compact (i.e. un fermé borné) non vide de \mathbb{R}^n et $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ une application continue sur X .*

Alors f est bornée et atteint ses bornes. Autrement dit, il existe $x \in X$ point de minimum global de f sur X i.e. :

$$\forall y \in X, f(x) \leq f(y).$$

Preuve. Introduisons l'image directe de X par f

$$f(X) = \{f(y) \mid y \in X\}.$$

Considérons une suite minimisante dans $f(X)$, c'est-à-dire une suite de $(x_n)_{n \in \mathbb{N}}$ d'éléments de X telle que

$$f(x_n) \longrightarrow \inf_{y \in X} f(y).$$

Comme X est fermé borné, il existe une sous-suite extraite $(x_{\sigma(n)})_{n \in \mathbb{N}}$ qui converge vers un $x \in X$. Cette suite extraite vérifie

$$x_{\sigma(n)} \longrightarrow x \quad \text{et} \quad f(x_{\sigma(n)}) \longrightarrow \inf_{y \in X} f(y).$$

Or f est continue, d'où par unicité de la limite, il suit

$$f(x) = \inf_{y \in X} f(y) \text{ avec } x \in X,$$

et f réalise son minimum sur X . □

Théorème 3.2 Soient F un fermé non vide de \mathbb{R}^n et $f : F \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ une application continue infinie à l'infini. Alors f admet un point de minimum sur F . Autrement dit, il existe $x \in F$ tel que

$$\forall y \in F, f(x) \leq f(y).$$

Preuve. (i) Définissons K compact. Comme $F \neq \emptyset$, nous avons : $\inf(F) \in \mathbb{R} \cup \{-\infty\}$. Soit $A \in \mathbb{R}$, tel que $A > \inf_{y \in F} f(y)$. Comme f est infinie à l'infini, il existe $R_1 > 0$ tel que pour $y \in \mathbb{R}^n$

$$\|y\| > R_1 \implies f(y) > A.$$

De plus F est non vide : il existe donc $R_2 > 0$ tel que

$$\overline{B}(0, R_2) \cap F \neq \emptyset.$$

Choisissons $R = \max(R_1, R_2)$

$$\|y\| > R \implies f(y) > A \quad \text{et} \quad \overline{B}(0, R) \cap F \neq \emptyset.$$

On introduit alors : $K = \overline{B}(0, R) \cap F$. L'ensemble K est un compact non vide car il est borné ($\|y\| \leq R$) et fermé (intersection de deux fermés).

(ii) Minimisons f sur K . Comme f est continue et K est compact, f atteint son minimum sur K , c'est-à-dire

$$\exists x \in K \quad | \quad f(x) = \inf_{y \in K} f(y), \tag{3.2}$$

(iii) Montrons que minimiser sur K revient à minimiser sur F . D'une part, nous avons

$$\inf_{y \in F} f(y) = \inf \left(\inf_{y \in K} f(y); \inf_{y \in F \setminus K} f(y) \right).$$

D'autre part, pour $z \in F$ et $z \notin K$, on a : $\|z\| \geq R$, soit :

$$f(z) > A > \inf_{y \in F} f(y)$$

Par conséquent : $\inf_{y \in F} f(y) < \inf_{y \in F \setminus K} f(y)$. Il suit

$$\inf_{y \in F} f(y) = \inf_{y \in K} f(y)$$

et d'après (3.2) il vient : $\exists x \in K \subset F \quad | \quad f(x) = \inf_{y \in F} f(y)$. □

3.2 Conditions d'optimalité

L'écriture des conditions d'optimalité en présence de contraintes est basée sur la même intuition que dans le cas sans contraintes, à savoir qu'il est impossible de descendre à partir d'un minimum. Toutefois les conditions d'optimalité vues au chapitre 2 ne sont plus valables comme le montre l'exemple suivant.

Exemple 3.2.1 Soit le problème :

$$\text{Minimiser } f(x) = x^2, x \in \mathbb{R}^n, \text{ sous la contrainte : } x \geq 1.$$

La solution de ce problème est : $x = 1$, et pourtant : $f'(1) = 2 \neq 0$.

Dans le cas où le domaine des contraintes est convexe, on a la condition nécessaire d'optimalité locale suivante :

Théorème 3.3 (Condition nécessaire d'optimalité locale) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable et X un convexe fermé, non vide de \mathbb{R}^n . Soit $x^* \in \mathbb{R}^n$ un point de minimum local du problème :

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous la contrainte : } x \in X.$$

Alors, pour tout $x \in X$, $\nabla f(x^*)^\top (x - x^*) \geq 0$.

Preuve. Soit $x \in X$. Par convexité de X , $x^* + \alpha(x - x^*)$ est un élément de X pour tout $\alpha \in [0, 1]$, d'où :

$$\forall \alpha \in [0, 1], f(x^* + \alpha(x - x^*)) \geq f(x^*)$$

puisque x^* est un point de minimum local de f sur X . On divise ensuite par $\alpha (> 0)$ et on fait tendre α vers 0^+ . \square

Rappelons que dans le cas convexe (i.e. si f est en plus convexe) alors la condition précédente devient nécessaire et suffisante.

Une autre approche : notion de direction admissible Une difficulté importante en optimisation sous contrainte consiste à savoir se déplacer dans l'ensemble des contraintes, i.e. étant donnée une direction de recherche comment garantir que l'on reste dans l'ensemble C . Pour cela, on introduit la notion de direction admissible.

Définition 3.1 (Direction admissible) Soit $x \in \mathbb{R}^n$ un point admissible du problème (3.1).

Une direction $d \in \mathbb{R}^n$ sera dite admissible en x s'il existe $\eta > 0$ tel que $x + sd$ soit admissible quel que soit $s \in]0, \eta]$.

Dans le cas particulier où le domaine des contraintes est convexe, déterminer une direction d admissible en x revient à déterminer un point admissible y , différent de x : $d = y - x$ est alors une direction admissible. En effet, quel que soit $\alpha \in [0, 1]$, $x + \alpha d = (1 - \alpha)x + \alpha y$ est une combinaison convexe d'éléments du convexe X , et donc un élément de X .

Autre preuve du théorème 3.3. Raisonnons par l'absurde. Supposons qu'il existe un point $x \in X$ tel que :

$$\nabla f(x^*)^\top (x - x^*) < 0.$$

D'après la définition 2.4 vue au chapitre 2, $d = x - x^*$ est donc une direction de descente de f au point x^* . En utilisant la caractérisation donnée par la proposition 2.3 : il existe $\eta > 0$ tel que :

$$\forall s \in]0, \eta], f(x^* + sd) < f(x^*).$$

Or : $x^* \in X$. De plus, $d = x - x^*$ définit une direction admissible du problème (3.1). Donc pour tout $s \in]0, \min(\eta, 1)]$, $x^* + sd$ est un point admissible de (3.1). □

3.3 Algorithme du gradient projeté

Construisons maintenant un algorithme de résolution du problème :

$$\text{Minimiser } f(x), x \in \mathbb{R}^n, \text{ sous la contrainte : } x \in X, \quad (3.3)$$

dans le cas où X est un sous-ensemble convexe fermé non vide de \mathbb{R}^n .

La méthode du gradient projeté s'inspire des méthodes de gradient décrites dans le chapitre précédent. L'idée de base consiste à suivre la direction de plus profonde descente, comme dans le cas sans contrainte :

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

où $s_k > 0$ est choisi de sorte que : $f(x_k + s_k d_k) < f(x_k)$. Toutefois, si $x_k \in X$, rien ne garantit que x_{k+1} appartienne également à X . Dès que l'on obtient un point non admissible, on projette celui-ci sur l'ensemble de contraintes X .

Projection sur un convexe. Soit X un convexe fermé, non vide de \mathbb{R}^n . La projection d'un point $x \in \mathbb{R}^n$ sur X , notée $p_X(x)$, est obtenue comme solution du problème d'optimisation suivant :

$$\text{Minimiser } \frac{1}{2} \|x - y\|_2^2 \quad \text{sous la contrainte : } y \in X \quad (3.4)$$

La fonctionnelle : $y \in \mathbb{R}^n \mapsto \frac{1}{2} \|x - y\|_2^2$ étant convexe et $\nabla f(y) = y - x$, le théorème 3.3 nous donne une condition nécessaire et suffisante pour que $x^* = p_X(x)$ soit solution de (3.4) :

$$\forall y \in X, (x^* - x)^\top (y - x^*) \geq 0. \quad (3.5)$$

On remarque en particulier que si $x \in X$, alors nécessairement : $x^* = x$.

Principe de l'algorithme. Soit x_k l'itéré courant. On génère à l'itération suivante le point admissible :

$$y_k = p_X(x_k - s_k \nabla f(x_k))$$

où p_X désigne l'opérateur de projection sur l'ensemble X . Avant d'aller plus loin, vérifions que la direction $d_k = y_k - x_k$, si elle est non nulle, est bien une direction de descente de f en x_k .

Lemme 3.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ supposée différentiable et $X \subset \mathbb{R}^n$ un convexe fermé, non vide. Notons x_k l'itéré courant et :

$$d(s) = p_X(x_k - s \nabla f(x_k)) - x_k, \quad s > 0$$

Si $d(s)$ est non nulle, alors $d(s)$ est une direction de descente pour tout $s > 0$.

Preuve. Soit $s > 0$ fixé. Supposons : $d(s) = p_X(x_k - s \nabla f(x_k)) - x_k \neq 0$. Il s'agit de démontrer que $d(s)$ est une direction de descente de f en x_k , autrement dit que $\nabla f(x_k)^\top d(s) < 0$.

D'après la caractérisation (3.5) de la projection sur un convexe, on peut écrire :

$$\forall y \in X, \left(p_X(x_k - s \nabla f(x_k)) - (x_k - s \nabla f(x_k)) \right)^\top \left(y - p_X(x_k - s \nabla f(x_k)) \right) \geq 0,$$

D'où, pour tout $y \in X$: $(d(s) + s \nabla f(x_k))^\top (y - x_k - d(s)) \geq 0$. Puisque $x_k \in X$, on choisit $y = x_k$, soit :

$$-(d(s) + s \nabla f(x_k))^\top d(s) \geq 0, \quad \text{ou encore : } \nabla f(x_k)^\top d(s) \leq -\frac{1}{s} d(s)^\top d(s) \leq 0.$$

Par hypothèse, $d(s) \neq 0$ ce qui implique : $\nabla f(x_k)^\top d(s) < 0$. □

Remarque 3.1 La direction $d(s)$ possède les propriétés suivantes :

1. Si $d(s) = 0$, alors : $p_X(x_k - s \nabla f(x_k)) = x_k$. Cela signifie simplement que la direction choisie par l'algorithme de gradient est orthogonale à l'ensemble X des contraintes en x_k . Le point x_k est alors un point stationnaire car la condition nécessaire d'optimalité (3.3) est satisfaite.
2. Supposons $d(s) \neq 0$. Alors x_k et $p_X(x_k - s \nabla f(x_k))$ sont des points admissibles du problème (3.3). La convexité de X nous garantit alors :

$$\forall \alpha \in [0, 1], x_k + \alpha d(s) \in X.$$

ALGORITHME DU GRADIENT PROJÉTÉ.

Données: f , p_X un opérateur de projection sur X , x_0 première approximation de la solution cherchée, $\varepsilon > 0$ précision demandée.

Sortie: une approximation x^* de la solution.

1. $k := 0$;
2. Tant que critère d'arrêt non satisfait,

- (a) *Projection sur X* : $y_k = p_X(x_k - s\nabla f(x_k))$
où s est le pas calculé par la méthode de gradient choisie ($s = 1$ par exemple) ;
- (b) *Direction de descente* : $d_k = y_k - x_k$;
- (c) *Recherche linéaire* : trouver un pas α_k tel que : $f(x_k + \alpha_k d_k) < f(x_k)$;
- (d) $x_{k+1} = x_k + \alpha_k d_k$; $k := k + 1$;

3. Retourner x_k .

Il est important de remarquer que le calcul à l'étape 2(a) du projeté sur X , peut parfois être aussi difficile que le problème initial. En effet y_k est obtenu en résolvant le problème :

$$\begin{aligned} \min_{y \in \mathbb{R}^n} \quad & \frac{1}{2} \|x_k - s\nabla f(x_k) - y\|_2^2 \\ \text{s.c.} \quad & x \in X. \end{aligned}$$

Il s'agit donc de résoudre un problème d'optimisation sur un convexe, avec une fonction objectif convexe. Lorsque le domaine X des contraintes est simple (contraintes de bornes en particulier), c'est faisable. Dès que les contraintes ne sont pas des contraintes de bornes, le calcul de la projection devient beaucoup plus délicat.

Exemple 3.3.1 *On veut résoudre par une méthode de gradient projeté le problème suivant :*

$$\begin{aligned} \min_{(x,y) \in \mathbb{R}^2} \quad & f(x,y) = \frac{1}{2}x^2 + \frac{7}{2}y^2 \\ \text{s.c.} \quad & -x + y = 1. \end{aligned}$$

Le domaine des contraintes $X = \{(x,y) \in \mathbb{R}^2 / -x + y = 1\}$ est un convexe fermé. Pour l'avoir résolu au chapitre 2, on sait que le problème hors contrainte admet un minimum global en $(0,0)$. Cependant $(0,0)$ ne satisfait pas la contrainte, ce n'est donc pas un point admissible du problème avec contrainte.

Afin de mettre en oeuvre l'algorithme de gradient projeté, il nous faut choisir les méthodes de calcul des pas s_k et α_k aux étapes 2(a) et 2(c) de notre algorithme :

- Étape 2(a) : on choisit une méthode de gradient pour le calcul du pas s_k .
- Étape 2(c) : en première approche, on choisit un pas fixe $\alpha_k = 1$, ce qui implique : $x_{k+1} = y_k = p_X(x_k - s\nabla f(x_k))$.

Le comportement numérique de la méthode de gradient projeté ainsi définie, est illustré par la figure 3.1 et le tableau 3.1 des itérations. On observe une convergence de la suite d'itérés générés à partir du point $x_0 = (4, 5.5)$ vers le point $(-0.875, 0.125)$.

Vérifions analytiquement ce résultat. On cherche un point $(x,y) \in \mathbb{R}^2$ vérifiant la contrainte : $y = 1 + x$ et minimisant f , ce qui revient à minimiser sans contrainte l'application :

$$\tilde{f} : x \mapsto f(x, 1+x) = 4x^2 + 7x + \frac{7}{2}.$$

Remarquons que \tilde{f} est strictement convexe ; d'après le théorème 2.3 appliqué au point $x^* = (-\frac{7}{8}, \frac{1}{8}) = (-0.875, 0.125)$, \tilde{f} admet un point de minimum global en x^* .

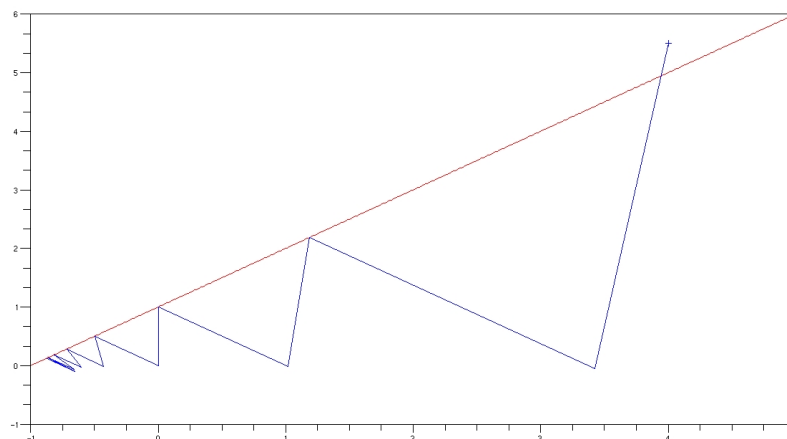


FIGURE 3.1 – Itérations successives (avant et après projection) de l'algorithme du gradient projeté - exemple 3.3.1 à partir du point (4, 5.5)

k	x_k	$x_k - s_k \nabla f(x_k)$	$\ d_k\ = \ y_k - x_k\ $
0	[4 5.5]	[3.4232925 - 0.0508095]	4.3472097
1	[1.1862415 2.1862415]	[1.0159064 - 0.0112495]	1.6743058
2	[0.0023285 1.0023285]	[0.0019958 - 9.462e - 08]	0.7089885
3	[-0.4990021 0.5009979]	[-0.4264826 - 0.0086691]	0.3091099
4	[-0.7175758 0.2824242]	[-0.6037028 - 0.0313035]	0.1413186
5	[-0.8175032 0.1824968]	[-0.6619890 - 0.0605186]	0.0618727
6	[-0.8612538 0.1387462]	[-0.6636631 - 0.0840740]	0.0178399
7	[-0.8738685 0.1261315]	[-0.6570769 - 0.0929058]	0.0015879
8	[-0.8749913 0.1250087]	[-0.6562565 - 0.0937435]	0.0000123
9	[-0.875 0.125]	[-0.65625 - 0.09375]	7.301e - 10

TABLE 3.1 – Itérations de la méthode du gradient projeté : on note $y_k = p_X(x_k - s_k \nabla f(x_k))$. Le critère d'arrêt $\|d_k\| < \varepsilon$ est satisfait en 10 itérations pour une précision $\varepsilon = 10^{-5}$.

Annexe A

Compléments

A.1 Rappels de calcul différentiel

A.1.1 Différentiabilité et gradient

Définition A.1 Soit $J : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonctionnelle. J est différentiable au point x ssi il existe $\nabla J(x) \in \mathbb{R}^n$ et $\mathbf{h} \mapsto \varepsilon(\mathbf{h}, x)$ vérifiant

$$J(x + \mathbf{h}) = J(x) + [\nabla J(x)]^\top \mathbf{h} + \|\mathbf{h}\| \varepsilon(\mathbf{h}, x), \quad \forall \mathbf{h} \in \mathbb{R}^n, \quad (\text{A.1})$$

avec

$$\varepsilon(\mathbf{h}, x) \xrightarrow{\mathbf{h} \rightarrow 0} 0. \quad (\text{A.2})$$

Lorsqu'il existe, le gradient peut être exprimé à l'aide des dérivées partielles

$$\nabla J(x) = \begin{bmatrix} \partial_{x_1} J(x) \\ \partial_{x_2} J(x) \\ \dots \\ \partial_{x_n} J(x) \end{bmatrix}. \quad (\text{A.3})$$

Définition A.2 Soit $J : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonctionnelle. J est de classe C^1 .

- J est différentiable au point x ,
- chacune des composantes de ∇J est continue.

A.1.2 Dérivées d'ordre supérieur

Définition A.3 On dit que $J : \mathbb{R}^n \rightarrow \mathbb{R}$ est deux fois différentiable au point $x \in \mathbb{R}^n$ ssi

- J est différentiable ;
- chacune des composantes de ∇J est différentiable.

Définition A.4 Soit $J : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable. La Hessienne de J définie à l'aide de ses dérivées partielles d'ordre 2

$$H[J](x) = [\partial_i \partial_j J(x)]_{1 \leq i, j \leq n}. \quad (\text{A.4})$$

Définition A.5 Une fonctionnelle $J : \mathbb{R}^n \rightarrow \mathbb{R}$ est de classe \mathcal{C}^2 ssi

- J est deux fois différentiable ;
- chacune des composantes de $H[J]$ est continue.

Théorème A.1 (de Schwarz) Soit $J : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonctionnelle de classe \mathcal{C}^2 .

La matrice $H[J](x), \forall x \in \mathbb{R}^n$, est symétrique.

A.1.3 Formules de Taylor

Pour une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$, les formules de Taylor s'écrivent

$$f(x+h) = f(x) + h f'(x) + \frac{h^2}{2} f''(x) + \dots + \frac{h^n}{n!} f^{(n)}(x) + \mathcal{O}(h^{n+1}). \quad (\text{A.5})$$

- La dérivée $f'(x)$ au point x définit le terme linéaire de ce développement, $h \rightarrow f'(x) h$.
- La dérivée $f''(x)$ définit le terme quadratique de ce développement, $h \rightarrow \frac{1}{2} h f''(x) h$.

$$\begin{array}{l} g(h) = \mathcal{O}(h^p) \Leftrightarrow \left\{ \begin{array}{l} \exists a > 0, \exists M > 0, |h| < a \Rightarrow \left| \frac{g(h)}{h^p} \right| \leq M : \\ \text{“}g(h) \text{ tend vers } 0 \text{ avec } h \text{ aussi vite que } h^p\text{“} . \end{array} \right. \\ g(h) = o(h^p) \Leftrightarrow \left\{ \begin{array}{l} \lim_{h \rightarrow 0} \frac{g(h)}{h^p} = 0, \\ \text{“}g(h) \text{ tend vers } 0 \text{ avec } h \text{ plus vite que } h^p\text{“} \end{array} \right. \end{array}$$

Pour une fonction $f : \mathbb{R}^p \rightarrow \mathbb{R}$, les formules de Taylor s'écrivent

$$\begin{aligned} f(x+h) &= f(x) + \sum_{j=1}^p h_j \partial_{x_j} f(x) + \frac{1}{2} \sum_{j=1}^p \sum_{k=1}^p h_j h_k \partial_{x_j x_k}^2 f(x) + \mathcal{O}(\|h^3\|), \\ &= f(x) + \sum_{j=1}^p h_j \partial_{x_j} f(x) + \frac{1}{2} \sum_{j=1}^p h_j \sum_{k=1}^p h_k \partial_{x_k x_j}^2 f(x) + \mathcal{O}(\|h^3\|), \end{aligned}$$

- $J_f(x)$ = Matrice Jacobienne = $[\partial_{x_1} f(x), \dots, \partial_{x_p} f(x)] = (\nabla f(x))^T$.
Le gradient de f est le vecteur transposé de sa matrice jacobienne.
- $H_f(x)$ = Matrice Hessienne = $\left[\partial_{x_i x_j}^2 f(x) \right]_{1 \leq i, j \leq p} \in M_p(\mathbb{R})$. Elle est symétrique.

$$f(x+h) = f(x) + J_f(x) h + \frac{1}{2} h^T H_f(x) h + \mathcal{O}(\|h^3\|). \quad (\text{A.6})$$

- La notation AB désigne le produit matriciel. Dans le cas du produit d'une matrice ligne par une matrice colonne, on peut utiliser une notation de produit scalaire de deux matrices (vecteurs) colonnes

$$(\text{A.6}) \Leftrightarrow f(x+h) = f(x) + \langle \nabla f(x), h \rangle + \frac{1}{2} \langle H_f(x) h, h \rangle + \mathcal{O}(\|h^3\|). \quad (\text{A.7})$$

Pour une fonction $f : \mathbb{R}^p \longrightarrow \mathbb{R}^n$, les formules de Taylor s'écrivent

$$f(x+h) = f(x) + L_f(h) + \frac{1}{2} Q_f(h) + \mathcal{O}(\|h^3\|). \quad (\text{A.8})$$

- $x, h \in \mathbb{R}^p$ et $f(x) \in \mathbb{R}^n$: (A.8) est une égalité dans \mathbb{R}^n . $L_f(h)$ désigne un terme (vecteur de \mathbb{R}^n) linéaire en h , $Q_f(h)$ désigne un terme quadratique en h . Pour les expliciter, on peut écrire une formule de type (A.6) pour chaque composante f_i de f ; pour $1 \leq i \leq n$,

$$f_i(x+h) = f_i(x) + J_{f_i}(x) h + \frac{1}{2} h^T H_{f_i}(x) h + \mathcal{O}(\|h^3\|).$$

- En rassemblant toutes les lignes et suivant les règles du produit matrice-vecteur, on obtient $L_f(h) = J_f(x) h$ avec

$$J_f(x) = \text{Matrice Jacobienne} = \begin{bmatrix} J_{f_1}(x) \\ \dots \\ J_{f_n}(x) \end{bmatrix} \in M_{n,p}(\mathbb{R}).$$

- Pour chaque i , le terme quadratique du développement de f_i est donné par $\frac{1}{2} h^T H_{f_i}(x) h$. En rassemblant toutes les lignes et en factorisant h à droite, on obtient :

$$\begin{bmatrix} h^T H_{f_1}(x) h \\ \dots \\ h^T H_{f_n}(x) h \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^p h_k \partial_{x_k x_1}^2 f_1(x) & \dots & \sum_{k=1}^p h_k \partial_{x_k x_p}^2 f_1(x) \\ \dots & \dots & \dots \\ \sum_{k=1}^p h_k \partial_{x_k x_1}^2 f_n(x) & \dots & \sum_{k=1}^p h_k \partial_{x_k x_p}^2 f_n(x) \end{bmatrix} \begin{bmatrix} h_1 \\ \dots \\ h_p \end{bmatrix}. \quad (\text{A.9})$$

- On a obtenu une expression de $Q_f(h)$ qui permet d'écrire (A.8) sous la forme

$$f(x+h) = f(x) + \begin{bmatrix} J_{f_1}(x) \\ \dots \\ J_{f_n}(x) \end{bmatrix} h + \frac{1}{2} \begin{bmatrix} h^T H_{f_1}(x) \\ \dots \\ h^T H_{f_n}(x) \end{bmatrix} h + \mathcal{O}(\|h^3\|).$$

A.2 Quelques démonstrations

A.2.1 Méthode de Gauss-Newton pour la résolution des problèmes de moindres carrés

Problème de moindres carrés

- $F : \mathbb{R}^p \longrightarrow \mathbb{R}^n, n > p : F(x) = \begin{bmatrix} F_1(x_1, \dots, x_p) \\ \dots \dots \\ F_n(x_1, \dots, x_p) \end{bmatrix}.$
- $r(x) = \frac{1}{2} \|F(x)\|^2 = \frac{1}{2} \langle F(x), F(x) \rangle.$
- On cherche $x^* = \operatorname{argmin} \{r(x)\} \Leftrightarrow r(x^*) = \min \left\{ \frac{1}{2} \sum_{i=1}^n F_i^2(x), x \in \mathbb{R}^p \right\}.$

Utilisation de la méthode de Newton

- Condition nécessaire d'optimalité du premier ordre : $\nabla r(x^*) = 0.$
- Résolution par la méthode de Newton : pour $k \geq 0,$

$$x^{(k+1)} = x^{(k)} - (H_r(x^{(k)}))^{-1} \nabla r(x^{(k)}). \quad (\text{A.10})$$

Calcul de $\nabla r(x)$: on peut le faire en **identifiant de la partie linéaire** d'un développement au voisinage de x :

$$\begin{aligned} 2 r(x+h) &= \langle F(x+h), F(x+h) \rangle, \\ &= \langle F(x) + J_F(x)h + \mathcal{O}(\|h\|^2), F(x) + J_F(x)h + \mathcal{O}(\|h\|^2) \rangle, \\ &= \langle F(x), F(x) \rangle + \langle F(x), J_F(x)h \rangle + \langle J_F(x)h, F(x) \rangle + \mathcal{O}(\|h\|^2). \end{aligned}$$

- Comme en dimension 1, on assimile les termes qui seront au moins $\mathcal{O}(\|h\|^2)$: $\langle F(x), \mathcal{O}(\|h\|^2) \rangle + \langle J_F(x)h, J_F(x)h \rangle + \dots = \mathcal{O}(\|h\|^2).$
- Pour $A \in M_{n,p}(\mathbb{R}), x \in \mathbb{R}^n$ et $y \in \mathbb{R}^p,$ on a $\langle x, Ay \rangle = \langle A^T x, y \rangle :$

$$r(x+h) = r(x) + \langle J_F(x)^T F(x), h \rangle + \mathcal{O}(\|h\|^2).$$

- Par identification,

$$\nabla r(x) = J_F(x)^T F(x). \quad (\text{A.11})$$

Calcul de $\nabla r(x)$: on peut le faire **en calculant les dérivées partielles** :

– Pour tout $i, 1 \leq i \leq p$,

$$\partial_{x_i} r(x) = \partial_{x_i} \left(\frac{1}{2} \sum_{j=1}^n F_j^2(x) \right) = \sum_{j=1}^n (\partial_{x_i} F_j(x)) F_j(x).$$

– En prenant en compte toute les lignes,

$$\begin{bmatrix} \partial_{x_1} r(x) \\ \dots \\ \partial_{x_p} r(x) \end{bmatrix} = \begin{bmatrix} \partial_{x_1} F_1(x) & \dots & \partial_{x_1} F_n(x) \\ \dots & \dots & \dots \\ \partial_{x_p} F_1(x) & \dots & \partial_{x_p} F_n(x) \end{bmatrix} \begin{bmatrix} F_1(x) \\ \dots \\ F_n(x) \end{bmatrix} = \sum_{j=1}^n (\nabla F_j) F_j = J_F^T F.$$

Calcul de $H_r(x)$: on peut le faire par **identification de la partie quadratique** d'un développement au voisinage de x :

$$\begin{aligned} r(x+h) &= \frac{1}{2} \left\langle F(x) + J_F(x)h + \frac{1}{2}Q_r(h) + \mathcal{O}(\|h\|^3), F(x) + J_F(x)h + \frac{1}{2}Q_F(h) + \mathcal{O}(\|h\|^3) \right\rangle, \\ &= r(x) + \langle \nabla r(x), h \rangle + \frac{1}{2} \langle J_F(x)h, J_F(x)h \rangle \dots \\ &\dots + \frac{1}{4} \langle F(x), Q_F(h) \rangle + \frac{1}{4} \langle Q_F(h), F(x) \rangle + \mathcal{O}(\|h\|^3), \end{aligned} \tag{A.12}$$

– La partie quadratique de (A.12) s'identifie selon

$$\langle H_r(x)h, h \rangle = \langle J_F(x)^T J_F(x)h, h \rangle + \frac{1}{2} \langle F(x), Q_F(h) \rangle$$

– Suivant (A.9), $\langle F(x), Q_F(h) \rangle = \left(\sum_{i=1}^n F_i(x) h^T H_{F_i}(x) \right) h$, d'où

$$H_r = J_F^T J_F + \sum_{i=1}^n F_i H_{F_i}. \tag{A.13}$$

Calcul de $H_r(x)$: on peut le faire **en dérivant $\nabla r(x)$** :

– $\nabla r = \sum_{j=1}^n (\nabla F_j) F_j$.

– On utilise la règle de dérivation d'un produit : $(\nabla r)' = \sum_{j=1}^n (H_{F_j} F_j + (\nabla F_j) J_{F_j})$.

– $J_{F_j} = (\nabla F_j)^T$, et $\sum_{j=1}^n (\nabla F_j) (\nabla F_j)^T = [\nabla F_1 \mid \dots \mid \nabla F_n] \begin{bmatrix} \nabla F_1^T \\ \dots \\ \nabla F_n^T \end{bmatrix} = J_F^T J_F$.

Calcul de $H_r(x)$: on peut **laborieusement** le faire en calculant les dérivées partielles secondes ; on ne le fait pas.