



Département Sciences du Numérique

Calcul différentiel - Optimisation sans contraintes - Premiers algorithmes

O. Cots, J. Gergaud, S. Gratton, D. Ruiz et E. Simon

14 octobre 2020

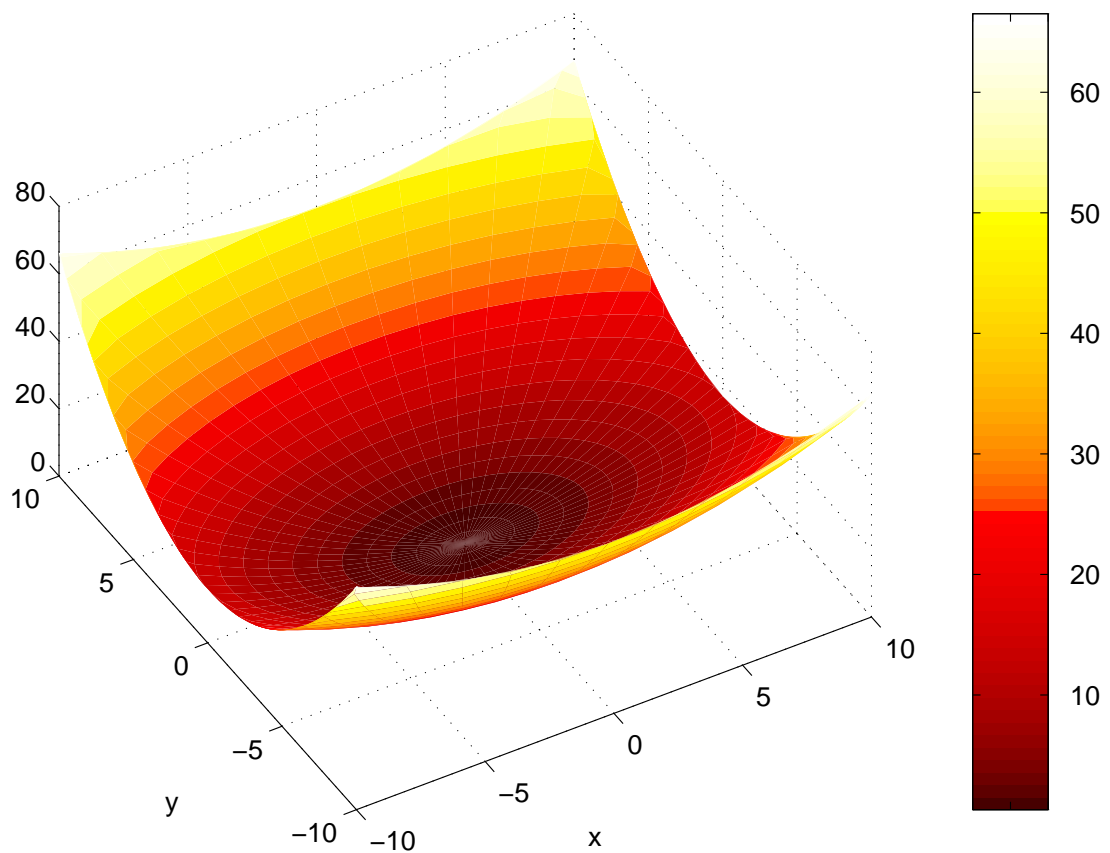


Table des matières

Chapitre 1. Exemples et définitions	3
1.1 Exemples	3
1.1.1 Cas continu et de dimension finie	3
1.1.2 Problèmes en nombres entiers	11
1.1.3 Problème en dimension infinie	13
1.2 Problème d'optimisation	15
1.2.1 Définitions	15
1.2.2 Classification	19
1.3 Exercices	20
Chapitre 2. Formes bilinéaires et quadratiques	22
2.1 Forme bilinéaire – Matrice d'une forme bilinéaire	22
2.1.1 Formes bilinéaires	22
2.1.2 Représentation matricielle d'une forme bilinéaire	22
2.1.3 Exemple dans \mathbb{R}^3	23
2.2 Formes quadratiques	23
2.2.1 Propriétés	24
2.3 Formes quadratiques définies positives	25
2.3.1 Produit scalaire	25
2.3.2 Exemples	25
2.4 Diagonalisation des endomorphismes symétriques	26
2.4.1 Introduction	26
2.4.2 Généralisation	27
2.5 Diagonalisation d'une forme quadratique	27
2.6 Compléments	29
Chapitre 3. Différentiabilité – Convexité	33
3.1 Dérivées de fonctions à plusieurs variables	33
3.1.1 Dérivée première	33
3.1.2 Dérivée seconde	34
3.1.3 Formule des accroissements finis - Formules de Taylor	35
3.1.4 Dimension finie et dérivées partielles	36
3.2 Convexité des fonctionnelles	39
3.2.1 Ensembles convexes - fonctionnelles convexes	39
3.2.2 Convexité et dérivée première	40
3.2.3 Convexité et dérivée seconde	41
Chapitre 4. Existence de solution, unicité de solution	43
4.1 Introduction	43
4.2 Existence de solution	43
4.2.1 Problèmes avec contraintes	43
4.2.2 Problème sans contraintes	44
4.3 Cas convexe	45

Chapitre 5. Condition nécessaire, condition suffisante de solution	
Cas sans contraintes et cas de contraintes convexes	47
5.1 Condition du premier ordre	47
5.1.1 Cas sans contraintes	47
5.1.2 Cas de contraintes convexes	47
5.1.3 Problèmes convexes	48
5.2 Conditions du deuxième ordre	48
5.2.1 Condition nécessaire	48
5.2.2 Condition suffisante	49
5.3 Exercices	50
Chapitre 6. Problèmes aux moindres carrés	51
6.1 Introduction	51
6.2 Les moindres carrés linéaires	51
6.2.1 Rappels	51
6.2.2 Application : approximation d'une fonction au sens des moindres carrés	52
6.3 La méthode de Newton	53
6.3.1 Introduction	53
6.3.2 Algorithme de Newton pour résoudre $f(x) = 0$	54
6.3.3 Résolution d'équations : cas de la dimension n	55
6.3.4 Convergence	56
6.3.5 Application aux problèmes d'optimisation	57
6.4 Résolution des problèmes aux moindres carrés non linéaires	57
6.4.1 Algorithme de Newton	57
6.4.2 Algorithme de Gauß-Newton	58
6.4.3 Exemples	58
Bibliographie	61

Problèmes aux moindres carrés

6.1 Introduction

L'objectif de ce chapitre est de résoudre les problèmes aux moindres carrés non-linéaires.

$$(P_1) \begin{cases} \min f(\beta) = \frac{1}{2} \|r(\beta)\|^2 \\ \beta \in \mathbb{R}^p. \end{cases}$$

Dans le cas où l'application r est affine, le problème est linéaire et sera résolu facilement. Si r n'est pas affine, il nous faut développer des algorithmes pour calculer une solution. La branche des mathématiques qui s'occupe de ces questions pour les problèmes d'optimisation continues s'appelle l'optimisation continue et sera abordée en deuxième année. Nous allons ici nous limiter à des algorithmes de base pour les problèmes aux moindres carrés, qui sont l'algorithme de Newton et l'algorithme de Gauß-Newton.

6.2 Les moindres carrés linéaires

6.2.1 Rappels

Considérons le problème aux moindres carrés linéaire

$$(P2) \begin{cases} \min f(\beta) = \frac{1}{2} \|y - X\beta\|^2 \\ \beta \in \mathbb{R}^p. \end{cases}$$

Ce problème admet une solution. En effet, ce problème est équivalent à résoudre

$$(P3) \begin{cases} \min g(\gamma) = \frac{1}{2} \|y - \gamma\|^2 \\ \gamma \in \text{Im } X \subset \mathbb{R}^n. \end{cases}$$

Comme $\text{Im } X$ est un fermé et que g est 0-coercive, le théorème 4.4.2.2 démontre l'existence d'une solution.

Remarque 6.2.1. Le problème (P3) est en fait le problème de la projection orthogonale du vecteur y sur $\text{Im } X$. Il possède une unique solution car g est strictement convexe. Par contre le problème initial (P2) possède une ou une infinité de solutions suivant que le rang de X est p ou est strictement inférieur à p .

Le problème (P2) est un problème convexe et différentiable, par suite une solution est caractérisée par la condition nécessaire d'ordre 1, qui conduit au système d'équations suivant, aussi appelé dans ce cas *équations normales* :

$$\nabla f(\beta) = X^T X \beta - X^T y = 0. \quad (6.1)$$

Remarque 6.2.2. Considérons ici la matrice X comme l'expression d'une application linéaire de \mathbb{R}^p à valeurs dans \mathbb{R}^n . on a alors

$$\begin{aligned} \mathbb{R}^p &= \text{Ker } X^\perp \oplus \text{Ker } X & \mathbb{R}^n &= \text{Im } X \oplus \text{Im } X^\perp = \text{Im } X \oplus \text{Ker}(X^T) \\ \beta^* &= X^+ y & \gamma^* &= \text{Proj}_{\text{Im } X}(y) \\ X^+ X &= \text{Proj}_{(\text{Ker } X)^\perp} & X X^+ &= \text{Proj}_{\text{Im } X} \\ \text{Si } \text{rank}(X) &= p, \text{ alors } \text{Ker } X &= \{\vec{0}\} \text{ et } X^+ &= (X^T X)^{-1} X^T \end{aligned}$$

6.2.2 Application : approximation d'une fonction au sens des moindres carrés

Le problème de l'approximation d'une fonction f sur un intervalle $I \subset \mathbb{R}$ est fondamentalement différent de celui de l'interpolation. Il consiste à remplacer la fonction f considérée par une autre fonction $\mathcal{P}(x)$ (en général plus régulière, et facile à manipuler) de sorte que la distance entre f et \mathcal{P} soit aussi petite que possible.

On peut chercher par exemple un polynôme de bas degré, qui approche la fonction f en un sens à préciser sur l'intervalle I , ce qui diffère du problème d'interpolation qui consiste à trouver un polynôme de degré en général élevé qui coïncide au maximum avec la fonction f .

La notion de distance entre les fonctions f et \mathcal{P} est bien évidemment fondamentale dans la définition du procédé d'approximation. On pourra par exemple distinguer :

- (i) **l'approximation au sens de la convergence uniforme**, où il s'agit de minimiser

$$\max_{x \in I} |f(x) - \mathcal{P}(x)| = \|f - \mathcal{P}\|_{\infty} ,$$

(problème ne relevant pas des moindres carrés)

- (ii) **l'approximation en moyenne quadratique** où il s'agit de minimiser la quantité

$$\int_I (f(x) - \mathcal{P}(x))^2 dx = \|f - \mathcal{P}\|_{L^2(I)}^2 ,$$

- (iii) **l'approximation au sens des moindres carrés discrets**, utile lorsque f n'est connue que de manière discrète (c'est à dire sur un ensemble fini de points $x_i \in I$, $1 \leq i \leq m$) ; cette approximation consiste alors à minimiser la quantité

$$\sum_{i=1}^m (f(x_i) - \mathcal{P}(x_i))^2 .$$

6.2.2.1 Approximation en moyenne quadratique

On se propose d'approcher f sur l'intervalle $I = [a, b]$ par une fonction $\mathcal{P}(x)$, où \mathcal{P} est une combinaison linéaire d'un ensemble de n fonctions données $u_j \in L^2(I)$, $j = 1, \dots, n$,

$$\mathcal{P}(x) = \sum_{j=1}^n \beta_j u_j(x) ,$$

les coefficients β_j étant donc les inconnues à déterminer de façon à minimiser la quantité

$$f(\beta_1, \dots, \beta_n) = \|f - \mathcal{P}\|_{L^2(I)}^2 = \int_a^b (f(x) - \sum_{j=1}^n \beta_j u_j(x))^2 dx .$$

Remarque : Si on souhaite réaliser une approximation polynômiale de f , il suffira de choisir

$$u_j(x) = x^{j-1} , \quad j = 1, 2, \dots, n .$$

Proposition 6.2.1

Soit $\beta \in \mathbb{R}^n$ le vecteur des coefficients β_j , $j = 1, \dots, n$. Une condition nécessaire et suffisante pour que β réalise le minimum de la fonctionnelle $f(\beta)$, est que β soit solution du système linéaire :

$$\sum_{j=1}^n (u_j | u_i) \beta_j = (f | u_i) , \quad i = 1, \dots, n ,$$

avec

$$(u_j|u_i) = \int_a^b u_j(x) u_i(x) dx \quad \text{et} \quad (f|u_i) = \int_a^b f(x) u_i(x) dx.$$

6.2.2.2 Approximation au sens des moindres carrés discrets

Soit f une fonction dont on connaît la valeur sur un sous ensemble fini de points x_1, \dots, x_m . On se propose alors d'approcher la fonction f par une fonction \mathcal{P} de telle façon que la quantité

$$\sum_{i=1}^m (f(x_i) - \mathcal{P}(x_i))^2,$$

soit minimale.

De plus, on cherche \mathcal{P} sous la forme

$$\mathcal{P}(x) = \sum_{j=1}^n \beta_j u_j(x),$$

où les u_j sont des fonctions connues et les $\{\beta_j\}_{j=1}^n$ sont les inconnues à déterminer. On suppose de plus que $m > n$, pour ne pas être ramené à un problème d'interpolation. Ce problème correspond au *lissage d'une fonction f donnée par une combinaison linéaire de fonctions quelconques*.

Introduisons alors la matrice X à m lignes et n colonnes, ainsi que les vecteurs $\beta \in \mathbb{R}^n$ et $y \in \mathbb{R}^m$ définis par

$$X = \begin{pmatrix} u_1(x_1) & \cdots & u_n(x_1) \\ u_1(x_2) & \cdots & u_n(x_2) \\ \vdots & & \vdots \\ u_1(x_m) & \cdots & u_n(x_m) \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}, \quad \text{et} \quad y = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_m) \end{pmatrix}.$$

En utilisant ces notations, on a alors le résultat suivant :

Proposition 6.2.2

Il est facile de voir que le problème d'approximation décrit ci-dessus se ramène en fait à minimiser par rapport à $\beta \in \mathbb{R}^n$ la quantité

$$f(\beta) = \|X\beta - y\|_2^2,$$

et donc que le vecteur de coefficients $\beta \in \mathbb{R}^n$ recherché correspond à la solution au sens des moindres carrés du système linéaire surdéterminé

$$X\beta = y.$$

De plus, si les points x_i , $i = 1, \dots, m$, et les fonctions u_j , $j = 1, \dots, n$, sont choisis de telle façon que la matrice X soit de rang maximal, alors le problème d'approximation au sens des moindres carrés discrets précédent admet une solution unique $\beta \in \mathbb{R}^n$, solution du système linéaire

$$X^T X \beta = X^T y.$$

6.3 La méthode de Newton

6.3.1 Introduction

L'algorithme de Newton est à la base des algorithmes d'optimisation implémentés dans les bibliothèques numériques. Mais, avant de considérer le cas d'un problème d'optimisation, nous allons nous intéresser au problème de la résolution d'un système d'équations non linéaires à n équations et n inconnues.

6.3.2 Algorithme de Newton pour résoudre $f(x) = 0$

6.3.2.1 Résolution d'une équation : cas de la dimension 1

le problème est ici de résoudre numériquement une équation $f(x) = 0$ où la fonction f est une fonction réelle de la variable réelle. nous supposons de plus que cette fonction est dérivable. on considère alors l'algorithme suivant :

Algorithme 6.1.

Initialisation :

choisir $x^{(0)} \in \mathbb{R}$

choisir $\varepsilon > 0$ et $MaxIter$

$k := 0$

Corps :

répéter

Résoudre $f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)}) = 0$, soit $x^{(k+1)}$ la solution

$k := k + 1$

jusqu'à $(|f(x^{(k)})| < \varepsilon(|f(x^{(0)}| + 1))$ ou $(k = MaxIter)$

Remarque 6.3.1. (i) ici $f'(x^{(k)})$ appartient à \mathbb{R} .

(ii) l'algorithme peut se “bloquer” si $f'(x^{(k)}) = 0$, et donc dans ce cas l'algorithme ne fournit pas de solution.

(iii) cet algorithme ne converge pas toujours.

Illustration graphique 6.3.2. l'intersection de la tangente à f en $x^{(k)}$ avec l'axe des abscisses (cf. figure 6.1) est donnée par la solution de $f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)}) = 0$, soit :

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

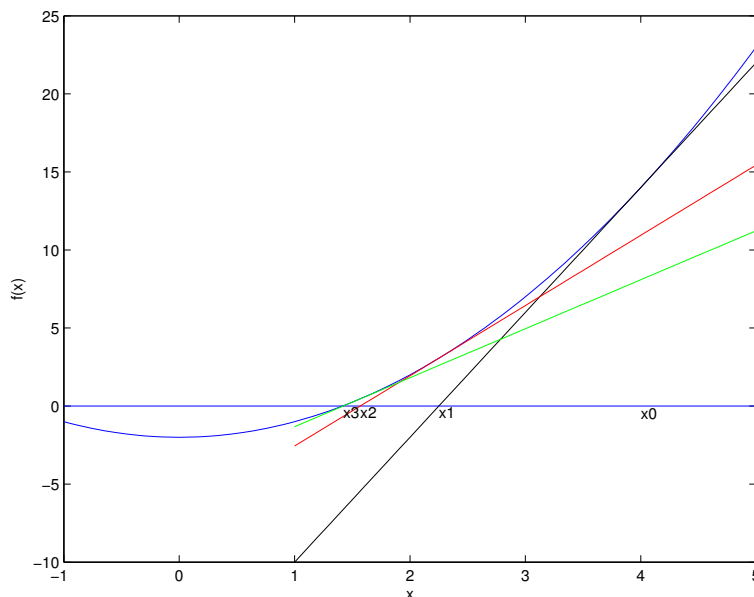


FIGURE 6.1 – Algorithme de Newton.

6.3.3 Résolution d'équations : cas de la dimension n

De la même façon qu'en dimension 1 on obtient l'algorithme en calculant $x^{(k+1)}$ à partir de $x^{(k)}$ donné en annulant la "meilleure" approximation affine de la fonction f au voisinage de $x^{(k)}$, c'est-à-dire en résolvant le système linéaire à n équations et à n inconnues suivant :

$$f(x^{(k)}) + J_f(x^{(k)})(x - x^{(k)}) = 0$$

qui admet une unique solution si $J_f(x^{(k)})$ est inversible.

Remarque 6.3.3. On présente très souvent l'algorithme de Newton sous la forme de la mise à jour du point courant donnée par 6.2. Cette équation est très utile pour la théorie, mais est bien évidemment à bannir pour une implémentation informatique.

$$x^{(k+1)} := x^{(k)} - [J_f(x^{(k)})]^{-1} f(x^{(k)}) \quad (6.2)$$

Remarque 6.3.4. lorsque $n = 1$, $f'(x^{(k)})$ est un réel et $[f'(x^{(k)})]^{-1} = 1/f'(x^{(k)})$, et nous retrouvons la mise à jour exposée précédemment.

en conclusion nous obtenons l'algorithme suivant :

Algorithme 6.2. [algorithme de newton]

Initialisation :

choisir $x^{(0)} \in \mathbb{R}^n$

choisir $\varepsilon > 0$ et $MaxIter$

$k := 0$

Corps :

répéter

Résoudre $f(x^{(k)}) + J_f(x^{(k)})(x - x^{(k)}) = 0$, soit $x^{(k+1)}$ la solution

$k := k + 1$

jusqu'à $(\|f(x^{(k)})\| < \varepsilon(\|f(x^{(0)})\| + 1))$ ou $(k = MaxIter)$

Remarque 6.3.5. (i) cet algorithme se "bloque" si $J_f(x^{(k)})$ n'est pas inversible.

(ii) le test d'arrêt $\|f(x^{(k)})\| < \varepsilon\|f(x^{(0)})\|$ signifie en fait que toutes les composantes de $f(x^{(k)})$ sont "proches" de 0 en relatif.

 **Exercice 6.3.1.** On considère la fonction

$$\begin{aligned} \text{soit } f : \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\longmapsto x^2 - a \text{ avec } a > 0. \end{aligned}$$

1. Donner l'itération de Newton pour résoudre $f(x) = 0$. □

 **Exercice 6.3.2.** On considère la fonction

$$\begin{aligned} f : \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &\longmapsto \begin{pmatrix} x_1 + x_2 - 3 \\ x_1^2 + x_2^2 - 9 \end{pmatrix} \end{aligned}$$

alors $f(x) = 0$ si et seulement si $x = (0, 3)^T$ ou $x = (3, 0)^T$.

1. Appliquer l'algorithme de Newton (6.2) en partant du point $x^{(0)} = (1, 5)^T$ et en prenant $\varepsilon = 0,6$. □

6.3.4 Convergence

Théorème 6.3.1

Soit f une fonction définie sur un ouvert Ω de \mathbb{R}^n à valeurs dans \mathbb{R}^n de classe C^2 dans $B(x^*, r) \subset \Omega$ et x^* un point de Ω tel que $f(x^*) = 0$. On suppose que $f'(x^*)$ est inversible, alors il existe $\varepsilon > 0$ tel que pour tout point $x^{(0)} \in B(x^*, \varepsilon)$, l'algorithme de Newton est bien défini et la suite des itérés $(x_k)_k$ converge vers x^* . De plus la convergence est quadratique, c'est-à-dire qu'il existe $c > 0$ tel que

$$\|x^{(k+1)} - x^*\| \leq c \|x^{(k)} - x^*\|^2 \quad (6.3)$$

Avant de voir la démonstration, voyons ce que signifie l'équation (6.3) (exemple provenant de [3] page 23). À la table (6.1) on voit que la convergence est plus rapide pour la fonction $f_1(x) = x^2 - 1$ qui vérifie les hypothèses du théorème que pour la fonction $f_2(x) = (x - 1)^2$ qui ne vérifie pas que $f'(1)$ soit inversible. La convergence quadratique signifie en pratique que si on est suffisamment près de la solution et si à une itération k on a p décimales qui sont exactes, on aura à l'itération $k + 1$, $2p$ décimales qui seront exactes.¹

	$f_1(x) = x^2 - 1$	$f_2(x) = (x - 1)^2$
x_0	2	2
x_1	1.25	1.5
x_2	1.025	1.25
x_3	1.0003048780488	1.125
x_4	1.0000000464611	1.0625
x_5	1.0	1.03125

TABLE 6.1 – Convergence quadratique pour f_1 et linéaire pour f_2 .

Démontrons maintenant le théorème.

► L'ensemble $\{x \in \Omega, J_f(x) \text{ inversible}\} = \mathbb{C}_\Omega\{x \in \Omega, \det \circ J_f(x) = 0\}$ est un ouvert car c'est le complémentaire de l'antécédent d'un fermé par une application continue. Par suite il existe $\varepsilon_1 > 0$ tel que pour tout $x \in B(x^*, \varepsilon_1)$, $J_f(x)$ soit inversible. Comme f est C^2 l'application qui à $x \in B(x^*, \varepsilon_1)$ associe $\|[J_f(x)]^{-1}\|$ est continue, on en déduit que pour $0 < \varepsilon_2 < \varepsilon_1$ l'image par cette application de $\overline{B(x^*, \varepsilon_2)}$ est un compact. Par suite, il existe $\beta > 0$ tel que pour tout $x \in \overline{B(x^*, \varepsilon_2)}$ on a $\|[J_f(x)]^{-1}\| \leq \beta$.

$$\|x^{(k+1)} - x^*\| = \|x^{(k)} - x^* - [J_f(x^{(k)})]^{-1} f(x^{(k)})\| \quad (6.4)$$

$$\leq \|[J_f(x^{(k)})]^{-1}\| \|f(x^{(k)}) - f(x^*) - J_f(x^{(k)})(x^{(k)} - x^*)\| \quad (6.5)$$

$$\leq \frac{1}{2} \|[J_f(x^{(k)})]^{-1}\| \sup_{x \in B_f(x^*, \varepsilon_2)} \|\nabla^2 f(x)\| \|x^{(k)} - x^*\|^2. \quad (6.6)$$

Mais f est C^2 par suite en posant $\sup_{x \in B_f(x^*, \varepsilon_2)} \|\nabla^2 f(x)\| = \gamma$ on obtient

$$\|x^{(k+1)} - x^*\| \leq \frac{1}{2} \beta \gamma \|x^{(k)} - x^*\|^2. \quad (6.7)$$

Posons maintenant $\varepsilon = \min(\varepsilon_2, 1/(\beta\gamma))$ et prenons $x^{(0)} \in B(x^*, \varepsilon)$, on obtient

$$\|x^{(1)} - x^*\| \leq \frac{1}{2} \|x^{(0)} - x^*\| \leq \varepsilon/2.$$

1. Ajouter le graphique, cf. Daniel.

Donc $x^{(1)} \in B(x^*, \varepsilon)$ et par récurrence

$$\|x^{(k)} - x^*\| \leq \frac{1}{2^k} \|x^{(0)} - x^*\|.$$

Par suite $x^{(k)} \in B(x^*, \varepsilon)$ et la suite des itérés de Newton existe et converge vers x^* . Quand à la convergence quadratique elle vient de l'inéquation (6.7). ■

6.3.5 Application aux problèmes d'optimisation

Nous rappelons que le problème qui nous intéresse ici est le suivant :

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

et nous avons vu (théorème (5.5.1.1)) qu'une condition nécessaire de solution est $f'(x) = 0$. Cette condition nous conduit donc tout naturellement vers la recherche d'un zéro de l'équation $g(x) = 0$ avec $g(x) = \nabla f(x)$. Il nous suffit donc d'appliquer l'algorithme de Newton à cette fonction g . A chaque itération nous aurons donc à résoudre le système linéaire suivant :

$$\nabla^2 f(x^{(k)})(x - x^{(k)}) + \nabla f(x^{(k)}) = 0. \quad (6.8)$$

Remarque 6.3.6. Si $\nabla^2 f(x^{(k)})$ n'est pas inversible alors l'algorithme se bloque.

Remarque 6.3.7. Nous avons vu dans la formule de Taylor-Young à l'ordre 2 (Chapitre 3, Théorème 3.1.3) que la meilleure approximation quadratique de la fonctionnelle f au voisinage du point $x^{(k)}$ est donnée par :

$$q(x) = f(x^{(k)}) + (\nabla f(x^{(k)})|x - x^{(k)}) + \frac{1}{2}(\nabla^2 f(x^{(k)})(x - x^{(k)})|x - x^{(k)})$$

et nous avons $\nabla q(x) = \nabla f(x^{(k)}) + \nabla^2 f(x^{(k)})(x - x^{(k)})$ et $\nabla^2 q(x) = \nabla^2 f(x^{(k)})$. Par suite si $\nabla^2 f(x^{(k)})$ est définie positive q est convexe et rechercher le minimum de $q(x)$ sur \mathbb{R}^n est équivalent à résoudre l'équation $\nabla q(x) = 0$. Mais cette dernière équation est équivalente à l'itération de Newton pour résoudre $\nabla f(x) = 0$. En conclusion notre algorithme recherche à chaque itération, lorsque $\nabla^2 f(x^{(k)})$ est définie positive, le minimum de l'approximation à l'ordre 2 de la fonctionnelle f .

6.4 Résolution des problèmes aux moindres carrés non linéaires

6.4.1 Algorithme de Newton

La fonction à optimiser s'écrit ici

$$f(\beta) = \frac{1}{2} \|r(\beta)\|^2 = \frac{1}{2} \sum_{i=1}^n r_i^2(\beta).$$

Nous avons donc

$$\begin{aligned} \nabla f(\beta) &= \sum_i r_i(\beta) \nabla r_i(\beta) = J_r(\beta)^T r(\beta) \\ \nabla^2 f(\beta) &= \sum_i r_i(\beta) \nabla^2 r_i(\beta) + \sum_i \nabla r_i(\beta) \nabla r_i(\beta)^T \\ &= S(\beta) + J_r(\beta)^T J_r(\beta) \end{aligned}$$

L'itération de l'algorithme de Newton s'écrit donc

$$\beta^{(k+1)} = \beta^{(k)} - [S(\beta^{(k)}) + J_r(\beta^{(k)})^T J_r(\beta^{(k)})]^{-1} J_r(\beta^{(k)})^T r(\beta^{(k)}) \quad (6.9)$$

6.4.2 Algorithme de Gauß-Newton

L'idée est ici de linéariser les résidus autour du point $\beta^{(k)}$ et ainsi de se ramener à un problème aux moindres carrés linéaire. Posons $s = \beta - \beta^{(k)}$, on cherche donc à chaque itération à résoudre

$$(P_k) \begin{cases} \min_{s \in \mathbb{R}^p} f_k(s) = \frac{1}{2} \|r(\beta^{(k)}) + J_r(\beta^{(k)})s\|^2 \end{cases}$$

ce qui est équivalent à résoudre les équations normales

$$J_r(\beta^{(k)})^T J_r(\beta^{(k)})s + J_r(\beta^{(k)})^T r(\beta^{(k)}) = 0.$$

Donc, si $J_r(\beta^{(k)})^T J_r(\beta^{(k)})$ est inversible, on peut écrire l'itération de Gauß-Newton :

$$\beta^{(k+1)} = \beta^{(k)} - [J_r(\beta^{(k)})^T J_r(\beta^{(k)})]^{-1} J_r(\beta^{(k)})^T r(\beta^{(k)}). \quad (6.10)$$

Remarque 6.4.1. (i) La différence entre les deux algorithmes réside dans l'absence du terme $S(\beta)$ dans l'équation (6.9), terme qui contient les matrices hessiennes des résidus.

(ii) Il y a deux avantages à l'algorithme de Gauß-Newton par rapport à l'algorithme de Newton :

- On n'a pas besoin de calculer les matrices hessiennes des résidus ;
- Contrairement à l'algorithme de Newton, on peut toujours trouver une solution $\beta^{(k)}$ à (P_k) .

6.4.3 Exemples

Exemple 6.4.1. Considérons le problème suivant

$$(P) \begin{cases} \min_{x \in \mathbb{R}^2} f(x) = \frac{1}{2}((x_1^2 - x_2)^2 + (1 - x_1)^2) \end{cases}$$

La figure 6.2 visualise la fonctionnelle à minimiser. Calculons le gradient de f et sa dérivée seconde :

$$\begin{aligned} \nabla f(x) &= \begin{pmatrix} 2x_1^3 - 2x_1x_2 + x_1 - 1 \\ -x_1^2 + x_2 \end{pmatrix}, \\ \nabla^2 f(x) &= \begin{pmatrix} 6x_1^2 - 2x_2 + 1 & -2x_1 \\ -2x_1 & 1 \end{pmatrix}. \end{aligned}$$

Prenons $x^{(0)} = (0 \ 1)^T$. À la première itération, nous avons à résoudre le système linéaire

$$\begin{aligned} \nabla^2 f(x^{(0)})(x - x^{(0)}) + \nabla f(x^{(0)}) &= 0 \iff \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 - 1 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &\iff \begin{cases} -x_1 + 0x_2 &= 1 \\ 0x_1 + x_2 &= 0 \end{cases} \end{aligned}$$

D'où

$$x^{(1)} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

À la deuxième itération, nous avons à résoudre le système linéaire

$$\begin{aligned} \nabla^2 f(x^{(1)})(x - x^{(1)}) + \nabla f(x^{(1)}) &= 0 \iff \begin{pmatrix} 7 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 + 1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -4 \\ -1 \end{pmatrix} \\ &\iff \begin{cases} 7x_1 + 2x_2 &= -3 \\ 2x_1 + x_2 &= -1 \end{cases} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{aligned}$$

D'où

$$x^{(2)} = \begin{pmatrix} -\frac{1}{3} \\ -\frac{1}{3} \end{pmatrix}.$$

La figure 6.3 donne les courbes de niveaux de la fonction à minimiser ainsi que les itérés successifs.

□

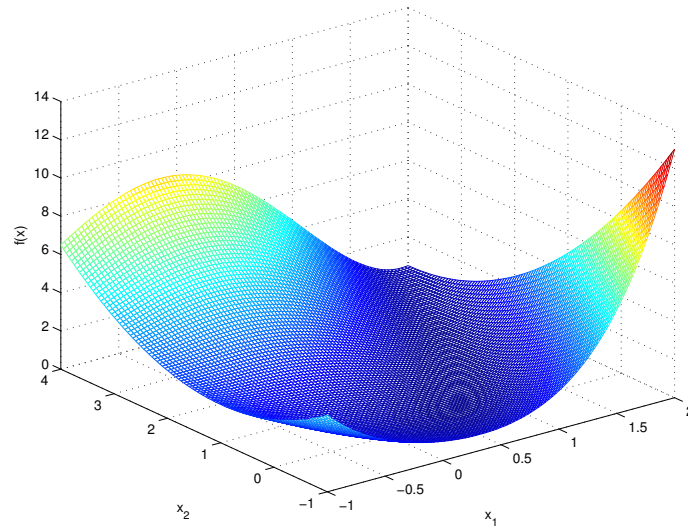
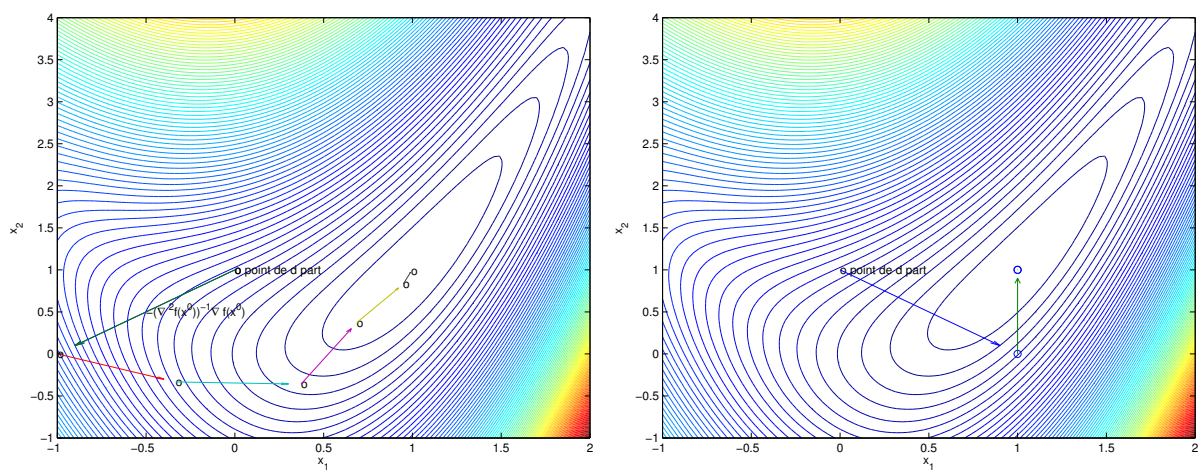
FIGURE 6.2 – Fonction f à minimiser pour l'exemple 6.4.1.

FIGURE 6.3 – Itérés de l'algorithme de Newton (Gauche), Gauß-Newton (Droite) pour l'exemple 6.4.1.

Bibliographie

- [1] Carpentier. Cours de 3ième année enseiht, filière informatique et mathématiques appliquées. INPT-ENSEEIHT, 1983. \leftrightarrow [12](#).
- [2] Gerard Cornuejols and Reha Tütüncü. *Optimization Methods in Finance*. Cambridge University Press, 2007. \leftrightarrow [11](#).
- [3] J.E. Dennis and Jr. Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1996. \leftrightarrow [56](#).
- [4] J.M. Devos. Fermat "le premier homme du monde". France 3, IREM de Toulouse, CRDP Midi-Pyrénées, 1995. Casette vidéo. \leftrightarrow [4](#).
- [5] Jean-Baptiste Hiriart-Urruty. *L'Optimisation*. Que sais-je. Presses Universitaires de France, 1996. ISBN : 2 13 047981 2. \leftrightarrow [50](#).