# *Introduction à l'optimisation : aspects théoriques, numériques et algorithmes*

**Xavier ANTOINE**[1,2,3] , **Pierre DREYFUSS**[2,3] et **Yannick PRIVAT**[2,3]

ENSMN-ENSEM 2A (2006-2007)

[1]Institut National Polytechnique de Lorraine (INPL), Ecole Nationale Supérieure d'Electricité et de Mécanique, Bureau 402, LORIA, 2 av. de la Forêt de Haye, BP 3 F-54501, Vandoeuvre-lès-Nancy, France.

[2]Ecole Nationale Supérieure des Mines de Nancy, Département de Génie Industriel, Parc de Saurupt, CS 14 234, 54042 Nancy cedex, France.

[3]Institut Elie Cartan Nancy (IECN), Université Henri Poincaré Nancy 1,B.P. 239, F-54506 Vandoeuvre-lès-Nancy Cedex, France.

# Table des matières

**Avant-propos**

Ce cours présente les bases de l'optimisation mathématique et numérique. Vous trouverez, lors des deux premiers chapitres, des rappels de base du calcul différentiel. Ces notions ont déjà été vues lors de votre cursus universitaire. Toutefois, afin de s'en assurer et pour avoir une notation et des notions auto-contenues, celles-ci sont détaillées. Dans le troisième chapitre, nous donnons quelques résultats théoriques sur l'optimisation sans puis avec contraintes. Ces développements sont destinés à mettre en place les notions utiles au développement d'algorithmes numériques. C'est le sujet du quatrième chapitre où nous introduisons les algorithmes classiques de l'optimisation numérique sans contrainte. Divers exercices et séances de travaux dirigés accompagnent le présent document afin d'assimiler les notions plus théoriques vues en cours. Les travaux dirigés sont développés pour être notamment implémentés sous le logiciel de calcul scientifique Matlab.

*X. ANTOINE, P. DREYFUSS & Y. PRIVAT*
*Nancy, le 19 juillet 2007*

# Chapitre 1

# Continuité et calcul différentiel de champs scalaires et vectoriels

## 1.1 Fonctions de $\mathbb{R}^n$ vers $\mathbb{R}^m$

Nous considérons ici des fonctions

$$f : V \to W$$

où $V$ et $W$ sont des espaces vectoriels de dimensions finies. Plus précisément, nous considérons le choix : $V = \mathbb{R}^n$ et $W = \mathbb{R}^m$. Lorsque $m = n = 1$, une telle fonction est appelée fonction d'une variable réelle à valeurs réelles. Lorsque $n = 1$ et $m > 1$, cette fonction est appelée une fonction vectorielle d'une variable réelle à valeurs réelles. Nous faisons l'hypothèse ici que $n > 1$ et $m \geq 1$. Lorsque $m = 1$, la fonction est appelée fonction à valeurs réelles d'une variable vectorielle réelle, ou plus brièvement, un *champ scalaire*. Lorsque $m > 1$, elle est appelée fonction à valeurs vectorielles réelle d'une variable vectorielle, ou tout simplement champ de vecteurs (réel).

Nous allons nous intéresser ici à étendre les concepts, connus, de limite, continuité, et dérivée à des champs scalaires et vectoriels. Nous utilisons, dans la suite du chapitre, les notations suivantes. Si $f$ est un champ scalaire défini en un point $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{R}^n$, les notations $f(\mathbf{x})$ et $f(x_1, ..., x_n)$ seront utilisées pour désigner la valeur de $f$ en ce point particulier. Si $\mathbf{f}$ est un champ de vecteurs, nous écrivons également $\mathbf{f}(\mathbf{x})$ ou $\mathbf{f}(x_1, ..., x_n)$.

Définissons le produit scalaire usuel de deux vecteurs réels comme

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_k y_k, \forall (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n,$$

et la norme associée

$$\|\mathbf{x}\| = (\mathbf{x} \cdot \mathbf{x})^{1/2}.$$

Les points dans le plan sont généralement notés $(x, y)$ à la place de $(x_1, x_2)$ et $(x, y, z)$ plutôt que $(x_1, x_2, x_3)$

dans le cas tridimensionnel.

Les champs scalaires et vectoriels définis sur des sous-ensembles de $\mathbb{R}^2$ ou $\mathbb{R}^3$ (voir plus) apparaissent très souvent et de manière naturelle dans les sciences de l'ingénieur. En effet, dans de nombreux problèmes, on s'intéresse aux variations d'un champ. Dans le cas unidimensionnel, c'est la dérivée qui traduit cette idée. La notion de dérivée s'applique aux fonctions définies sur des ouverts. Généralisons cette idée dans le cas de $\mathbb{R}^n$.

## 1.2 Notion de continuité

### 1.2.1 Boules ouvertes et ensembles ouverts

Soit $\mathbf{x}_0$ un point de $\mathbb{R}^n$ et $r > 0$ un nombre réel donné, strictement positif. L'ensemble des points $\mathbf{x}$ de $\mathbb{R}^n$ tels que : $\|\mathbf{x} - \mathbf{x}_0\| < r$, est appelé une $n$-boule ouverte de rayon $r$ et de centre $\mathbf{x}_0$. On la note $\mathcal{B}(\mathbf{x}_0; r)$. Un exemple est donné en dimension un par un intervalle ouvert de centre $\mathbf{x}_0$. Dans $\mathbb{R}^2$, nous retrouvons le disque circulaire ouvert de centre $\mathbf{x}_0$ et de rayon $r$. Dans $\mathbb{R}^3$, c'est la boule usuelle ouverte de centre $\mathbf{x}_0$ et de rayon $r$.

**Définition 1** *(d'un point intérieur et de l'intérieur de $\mathcal{S}$). Soit $\mathcal{S}$ un sous-ensemble de $\mathbb{R}^n$ et soit $\mathbf{x}_0 \in \mathcal{S}$. Alors, $\mathbf{x}_0$ est appelé un point intérieur de $\mathcal{S}$ si il existe une $n$-boule ouverte de centre $\mathbf{x}_0$, tous ses points appartenant à $\mathcal{S}$. L'ensemble de tous les points intérieurs de $\mathcal{S}$ est appelé l'intérieur de $\mathcal{S}$ est est noté* $\text{int}\mathcal{S}$.

Un ouvert contenant un point $\mathbf{x}_0$ est appelé un voisinage de $\mathbf{x}_0$.

**Définition 2** *(d'un ouvert). Un ensemble $\mathcal{S}$ de $\mathbb{R}^n$ est appelé ouvert si tous ses points sont des points intérieurs. En d'autres termes, si et seulement si $\mathcal{S} = \text{int}\mathcal{S}$.*

**Exemple 1** *En dimension un, nous pouvons donner l'exemple d'un intervalle ouvert, ou encore d'une réunion d'intervalles ouverts. Un contre-exemple est un intervalle fermé. En dimension deux, un disque ouvert est un exemple (sans compter le bord). Un autre exemple est un rectangle du type $]a, b[\times]c, d[$. Un contre-exemple est ou ouvert de $\mathbb{R}$ considéré dans $\mathbb{R}^2$.*

Introduisons maintenant la notion d'extérieur et de frontière.

**Définition 3** *(d'extérieur et de frontière). Un point $\mathbf{x}$ est dit être extérieur à un ensemble $\mathcal{S}$ dans $\mathbb{R}^n$ si il existe une $n$-boule $\mathcal{B}(\mathbf{x})$ ne contenant aucun point de $\mathcal{S}$. L'ensemble de tous les points dans $\mathbb{R}^n$ extérieurs à $\mathcal{S}$ est appelé l'extérieur de $\mathcal{S}$ et est noté* $\text{ext}\mathcal{S}$. *Un point qui n'est ni dans l'extérieur ou l'intérieur de $\mathcal{S}$ est appelé un point frontière de $\mathcal{S}$ et est noté $\partial\mathcal{S}$. Un ensemble $\mathcal{S}$ de $\mathbb{R}^n$ est dit fermé si son complémentaire dans $\mathbb{R}^n$ (noté $-\mathcal{S}$ ou encore $\mathcal{S}^c$) est ouvert.*

### 1.2.2 Limite et continuité de champs scalaires et vectoriels

Les concepts de limite et continuité sont facilement étendus à des champs scalaires et vectoriels. Nous allons reformuler ce concept pour des champs vectoriels, celui-ci étant directement applicable aux champs scalaires.

Avant cela, commençons par rappeler la définition de la limite puis continuité d'une fonction dans $\mathbb{R}$.

**Définition 4** *Soit $f$ une fonction de $\mathbb{R}$ dans $\mathbb{R}$. Nous dirons que la fonction $f$ admet comme limite $L$ en un point $x_0$ si*

$$\forall \varepsilon > 0, \exists \eta > 0, |x - x_0| < \eta \Rightarrow |L - f(x)| < \varepsilon.$$

Nous le noterons

$$\lim_{x \to x_0} f(x) = L.$$

**Définition 5** *Soit $f$ une fonction de $\mathbb{R}$ dans $\mathbb{R}$. La fonction $f$ est dite continue en $x_0$ si*

$$\forall \varepsilon > 0, \exists \eta > 0, |x - x_0| < \eta \Rightarrow |f(x_0) - f(x)| < \varepsilon.$$

Considérons maintenant une fonction $f : \mathcal{S} \to \mathbb{R}^m$, où $\mathcal{S}$ est un sous-ensemble de $\mathbb{R}^n$. Soient $\mathbf{x}_0 \in \mathbb{R}^n$ et $\mathbf{L} \in \mathbb{R}^m$. Nous écrivons

$$\lim_{\mathbf{x} \to \mathbf{x}_0} \mathbf{f}(\mathbf{x}) = \mathbf{L}, \tag{1.1}$$

ce qui signifie que

$$\lim_{\|\mathbf{x} - \mathbf{x}_0\| \to 0} \|\mathbf{f}(\mathbf{x}) - \mathbf{L}\| = 0. \tag{1.2}$$

Le symbole *limite* dans l'équation (1.2) est la limite au sens usuel du calcul élémentaire. Dans cette définition, il n'est pas nécessaire que $\mathbf{f}$ soit définie en $\mathbf{x}_0$. Ecrivons $\mathbf{h} = \mathbf{x} - \mathbf{x}_0$ ; alors, l'équation (1.2) devient

$$\lim_{\|\mathbf{h}\| \to 0} \|\mathbf{f}(\mathbf{x}_0 + \mathbf{h}) - \mathbf{L}\| = 0.$$

Pour des points dans $\mathbb{R}^2$, nous écrivons $(x, y)$ pour $\mathbf{x}$ et $(x_0, y_0)$ pour $\mathbf{x}_0$. Ainsi, la relation (1.1) prend la forme

$$\lim_{(x,y) \to (x_0,y_0)} \mathbf{f}(x, y) = \mathbf{L}.$$

Pour des points dans $\mathbb{R}^3$, nous considérons la notation $\mathbf{x} = (x, y, z)$ et $\mathbf{x}_0 = (x_0, y_0, z_0)$. Par conséquent, nous avons

$$\lim_{(x,y,z) \to (x_0,y_0,z_0)} \mathbf{f}(x, y, z) = \mathbf{L}.$$

Une fonction est dite continue en $\mathbf{x}_0$ si $\mathbf{f}$ est définie en $\mathbf{x}_0$ et si

$$\lim_{\mathbf{x} \to \mathbf{x}_0} \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0).$$

**Définition 6** *Nous dirons que $\mathbf{f}$ est continue en $\mathbf{x}_0$ sur un ensemble $\mathcal{S}$ si $\mathbf{f}$ est continue en tout point de $\mathcal{S}$. On le note $\mathbf{f} \in \mathcal{C}^0(\mathcal{S})$.*

Puisque ces définitions sont des extensions directes de celles établies dans le cas unidimensionnel, il n'est pas surprenant d'apprendre que beaucoup de propriétés familières de la limite et de la continuité peuvent aussi être étendues. Pour les champs scalaires, les théorèmes basiques concernant les limites et continuités de sommes, produits et quotients de champs scalaires peuvent être étendues directement. Pour les champs vectoriels, les quotients ne sont pas définis mais nous avons les théorèmes suivants.

**Théorème 1** *Si* $\lim_{\mathbf{x}\to\mathbf{x}_0} \mathbf{f}(\mathbf{x}) = \mathbf{L}$ *et* $\lim_{\mathbf{x}\to\mathbf{x}_0} \mathbf{g}(\mathbf{x}) = \mathbf{M}$*, nous avons également*

*a)* $\lim_{\mathbf{x}\to\mathbf{x}_0}(\mathbf{f}+\mathbf{g})(\mathbf{x}) = \mathbf{L}+\mathbf{M}$,

*b)* $\lim_{\mathbf{x}\to\mathbf{x}_0}(\lambda\mathbf{f})(\mathbf{x}) = \lambda\mathbf{L}$, $\forall\lambda\in\mathbb{R}$,

*c)* $\lim_{\mathbf{x}\to\mathbf{x}_0}(\mathbf{f}\cdot\mathbf{g})(\mathbf{x}) = \mathbf{L}\cdot\mathbf{M}$,

*d)* $\lim_{\mathbf{x}\to\mathbf{x}_0} \|\mathbf{f}(\mathbf{x})\| = \|\mathbf{L}\|$.

**Exemple 2** *(continuité des composantes d'un champ de vecteurs). Si un champ vectoriel* $\mathbf{f}$ *a ses valeurs dans* $\mathbb{R}^m$*, chaque valeur* $\mathbf{f}(\mathbf{x})$ *de la fonction a m composantes et nous pouvons écrire*

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_m(\mathbf{x})).$$

*Les m champs scalaires* $f_1$*,...,* $f_m$ *sont appelés composantes du champ de vecteur* $\mathbf{f}$*. On peut montrer que* $\mathbf{f}$ *est continue en un point si et seulement si chaque composante* $f_k$ *est continue en ce point.*

On peut construire d'autres exemples de fonctions continues grâce au théorème suivant sur les fonctions composées.

**Théorème 2** *Soient* $\mathbf{f}$ *et* $\mathbf{g}$ *des fonctions telles que la fonction composée* $\mathbf{f}\circ\mathbf{g}$ *soit définie en* $\mathbf{a}$*, où*

$$(\mathbf{f}\circ\mathbf{g})(\mathbf{x}) = \mathbf{f}(\mathbf{g}(\mathbf{x})).$$

*Si* $\mathbf{g}$ *est continue en* $\mathbf{x}_0$ *et si* $\mathbf{f}$ *est continue en* $\mathbf{g}(\mathbf{x}_0)$*, alors la composée* $\mathbf{f}\circ\mathbf{g}$ *est continue en* $\mathbf{x}_0$*.*

**Exemple 3** *Le théorème précédent implique la continuité des champs scalaires* $\mathbf{h}$*, où* $\mathbf{h}(x,y)$ *est donnée par les formules telles que*

*i)* $\sin(x^2y)$

*ii)* $\log(x^2+y^2)$

*iii)* $\dfrac{\exp(x+y)}{x+y}$

*iv)* $\log(\cos(x^2+y^2))$.

*Ces exemples conduisent à des fonctions continues en tout point où la fonction est définie. La première est continue en tous les points du plan, la seconde en tous les points en dehors de l'origine, la troisième en tous les points tels que* $x+y\neq 0$*, et, enfin, la quatrième en tous les points tels que* $x^2+y^2$ *n'est pas un multiple impair de* $\pi/2$*. Plus précisément, ce dernier ensemble correspond aux points* $(x,y)$ *tels que*

$$x^2+y^2 = \frac{\ell\pi}{2}, \ell = 1,3,5,...$$

*C'est une famille de cercles centrés à l'origine. Ces exemples montrent notamment que l'ensemble des discontinuités d'une fonction de deux variables peut être un ou des points isolés, des courbes entières ou des familles de courbes.*

**Exemple 4** *Une fonction de deux variables peut être continue en chacune des variables séparément et être discontinue comme une fonction de deux variables. Vous pouvez considérer à titre d'exemple la fonction définie*

*par*

$$f(x,y) = \frac{xy}{x^2+y^2}, \textit{ si } (x,y) \neq (0,0),$$
$$f(0,0) = 0.$$

## 1.3 Diverses notions de dérivations

Nous introduirons dans cette section plusieurs notions de dérivée d'un champ scalaire : dérivée par rapport à un vecteur, directionnelle, partielle, de Gâteaux et totale. Nous verrons que ces notions sont distinctes. Nous établirons cependant un certain nombre de liens entre elles et avec la propriété de continuité.
Dans l'exercice 1.5 nous proposons d'établir une synthèse plus complète.

### 1.3.1 La dérivée d'un champ scalaire par rapport à un vecteur

Soit $f$ un champ scalaire défini sur un ensemble $\mathcal{S}$ de $\mathbb{R}^n$, et soit $\mathbf{x}_0$ un point intérieur de $\mathcal{S}$. Nous souhaitons étudier la manière dont varie un champ scalaire lorsque nous bougeons de $\mathbf{x}_0$ vers un point proche. Par exemple, supposons que $f(\mathbf{x}_0)$ représente la température en un point $\mathbf{x}_0$ donné dans une salle chauffée dont une fenêtre est ouverte. Si nous nous rapprochons de la fenêtre, la température tend à décroître, si nous nous rapprochons du chauffage, la température augmente. En général, la manière dont le champ change dépend de la direction selon laquelle nous nous dirigeons à partir de $\mathbf{x}_0$.

Supposons que nous spécifions cette direction par un second vecteur $\mathbf{y}$. Plus précisément, supposons que nous allions de $\mathbf{x}_0$ vers $\mathbf{x}_0 + \mathbf{y}$ le long de la ligne joignant $\mathbf{x}_0$ et $\mathbf{x}_0 + \mathbf{y}$. Chaque point de ce segment est alors de la forme $\mathbf{x}_0 + h\mathbf{y}$, où $h \in \mathbb{R}$. La distance de $\mathbf{x}_0$ à $\mathbf{x}_0 + h\mathbf{y}$ est $\|h\mathbf{y}\| = |h| \, \|\mathbf{y}\|$. Puisque $\mathbf{x}_0$ est un point intérieur de $\mathcal{S}$, il existe une $n$-boule $\mathcal{B}(\mathbf{x}_0; r)$ contenue entièrement dans $\mathcal{S}$. Si $h$ est choisit tel que $|h| \, \|\mathbf{y}\| < r$, le segment reliant $\mathbf{x}_0$ à $\mathbf{x}_0 + \mathbf{y}$ se trouve dans $\mathcal{S}$. Supposons que $h \neq 0$ mais suffisamment petit pour garantir que $\mathbf{x}_0 + h\mathbf{h} \in \mathcal{S}$ et formons le quotient

$$\frac{f(\mathbf{x}_0 + h\mathbf{y}) - f(\mathbf{x}_0)}{h}. \tag{1.3}$$

Le numérateur de ce quotient nous dit comment varie la fonction lorsque nous bougeons de $\mathbf{x}_0$ à $\mathbf{x}_0 + h\mathbf{y}$. Le quotient est appelé le taux de variation de $f$ sur le segment de ligne joignant $\mathbf{x}_0$ à $\mathbf{x}_0 + h\mathbf{y}$. Intéressons nous au comportement de ce quotient lorsque $h \to 0$.

**Définition 7** *(de la dérivée d'un champ scalaire par rapport à un vecteur). Soit $f : \mathcal{S} \to \mathbb{R}$ un champ scalaire donné. Soit $\mathbf{x}_0$ un point intérieur de $\mathcal{S}$ et soit $\mathbf{y}$ un point arbitraire dans $\mathbb{R}^n$. La dérivée de $f$ en $\mathbf{x}_0$ par rapport à $\mathbf{y}$, notée $f'(\mathbf{x}_0; \mathbf{y})$, est définie par l'équation*

$$f'(\mathbf{x}_0; \mathbf{y}) = \lim_{h \to 0} \frac{f(\mathbf{x}_0 + h\mathbf{y}) - f(\mathbf{x}_0)}{h},$$

*lorsque la limite définie dans le membre de droite de l'équation ci-dessus existe.*

**Exercice 1** *Calculer $f'(\mathbf{x}_0; \mathbf{y})$ si la fonction $f$ est définie par $f(\mathbf{x}) = \|\mathbf{x}\|^2$, pour tout $\mathbf{x} \in \mathbb{R}^n$.*

Abordons maintenant le théorème de la valeur moyenne pour les champs scalaires.

**Théorème 3** *(de la valeur moyenne pour les dérivées de champs scalaires). Supposons que la dérivée* $f'(\mathbf{x}_0 + h\mathbf{y}; \mathbf{y})$ *existe pour tout* $h$ *dans l'intervalle* $0 \leq h \leq 1$*. Alors, pour un certain* $\theta$ *réel dans l'intervalle ouvert* $0 < \theta < 1$*, nous avons*

$$f(\mathbf{x}_0 + \mathbf{y}) - f(\mathbf{x}_0) = f'(\mathbf{z}; \mathbf{y}), \text{ où } \mathbf{z} = \mathbf{x}_0 + \theta\mathbf{y}.$$

### 1.3.2 Dérivées directionnelles, dérivées partielles et dérivée de Gâteaux

Dans le cas particulier où $\mathbf{y}$ est un vecteur unitaire (c'est à dire $\|\mathbf{y}\| = 1$), la distance entre $\mathbf{x}_0$ et $\mathbf{x}_0 + h\mathbf{y}$ est $|h|$. Dans ce cas, le quotient (1.3) représente le taux de variations de $f$ par unité de distance le long du segment joignant $\mathbf{x}_0$ à $\mathbf{x}_0 + h\mathbf{y}$ ; la dérivée $f'(\mathbf{x}_0; \mathbf{y})$ est appelée dérivée directionnelle.

**Définition 8** *(des dérivées directionnelles, partielles et de la dérivée de Gâteaux). Si* $\mathbf{y}$ *est un vecteur unitaire, la dérivée* $f'(\mathbf{x}_0; \mathbf{y})$ *est appelée dérivée directionnelle de* $f$ *en* $\mathbf{x}_0$ *selon la direction* $\mathbf{y}$*. En particulier, si* $\mathbf{y} = \mathbf{e}_k$ *(le* $k$*-ième vecteur unitaire des coordonnées), la dérivée directionnelle* $f'(\mathbf{x}_0; \mathbf{e}_k)$ *est appelée la dérivée partielle de* $f$ *par rapport à* $\mathbf{e}_k$ *et est également notée* $D_k f(\mathbf{x}_0)$*. Ainsi, nous avons*

$$D_k f(\mathbf{x}_0) = f'(\mathbf{x}_0; \mathbf{e}_k).$$

*La fonction* $f$ *est dite Gâteaux-dérivable en* $\mathbf{x}_0$ *si et seulement si* $f$ *est dérivable en* $\mathbf{x}_0$ *par rapport à toutes les directions* $\mathbf{y}$*, et que l'application* $\mathbf{y} \to f'(\mathbf{x}_0; \mathbf{y})$ *est linéraire. Cette dernière application linéaire est alors appelée dérivée de Gâteaux de* $f$ *en* $\mathbf{x}_0$

Les notations suivantes sont également utilisées pour les dérivées partielles en un point $\mathbf{a}$

$$D_k f(a_1, ..., a_n), \quad \frac{\partial f}{\partial x_k}(a_1, ..., a_n), \quad f'_{x_k}(a_1, ..., a_n).$$

Quelques fois, la dérivée $f'_{x_k}$ est écrite sans le prime comme $f_{x_k}$. Dans $\mathbb{R}^2$, les vecteurs unitaires des coordonnées s'écrivent $\mathbf{i}$ et $\mathbf{j}$. Si $\mathbf{x}_0 = (x_0, y_0)$, les dérivées partielles $f'(\mathbf{x}_0; \mathbf{i})$ et $f'(\mathbf{x}_0; \mathbf{j})$ s'écrivent aussi

$$\frac{\partial f}{\partial x}(x_0, y_0) \quad \text{et} \quad \frac{\partial f}{\partial y}(x_0, y_0),$$

respectivement. Dans $\mathbb{R}^3$, si $\mathbf{x}_0 = (x_0, y_0, z_0)$, les dérivées partielles $D_1 f(\mathbf{x}_0)$, $D_2 f(\mathbf{x}_0)$ et $D_3 f(\mathbf{x}_0)$ sont aussi notées

$$\frac{\partial f}{\partial x}(x_0, y_0, z_0), \quad \frac{\partial f}{\partial y}(x_0, y_0, z_0) \quad \text{et} \quad \frac{\partial f}{\partial z}(x_0, y_0, z_0).$$

### 1.3.3 Dérivées partielles d'ordre supérieur

Les dérivées partielles produisent de nouveaux champs scalaires $D_1 f$,..., $D_n f$ pour un champ $f$ donné. Les dérivées partielles de $D_1 f$,..., $D_n f$ sont appelées dérivées secondes de $f$. Pour les fonctions de deux variables, il y a quatre dérivées partielles secondes que l'on écrit

$$D_1(D_1 f) = \frac{\partial^2 f}{\partial x^2}, \quad D_1(D_2 f) = \frac{\partial^2 f}{\partial x \partial y}, \quad D_2(D_1 f) = \frac{\partial^2 f}{\partial y \partial x}, \quad \text{et} \quad D_2(D_2 f) = \frac{\partial^2 f}{\partial y^2}.$$

On utilise quelques fois la notation $D_{i,j}f$ pour la dérivée seconde $D_i(D_j f)$. Par exemple, $D_{1,2}f = D_1(D_2 f)$. Selon la notation $\partial$, on précise l'ordre de dérivation en écrivant

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}(\frac{\partial f}{\partial y}).$$

Il est possible que cette quantité soit égale ou non à l'autre dérivée mixte $\dfrac{\partial^2 f}{\partial y \partial x}$. Nous montrerons à la fin de ce chapitre que les deux dérivées mixtes sont égales en un point si au moins une d'entre elles est continue dans un voisinage de ce point. Nous donnerons également un contre-exemple.

**Exercice 2** *Un champ scalaire $f$ est défini sur $\mathbb{R}^n$ par l'équation $f(\mathbf{x}) = \mathbf{x}_0 \cdot \mathbf{x}$, où $\mathbf{x}_0$ est un vecteur constant.*

1. *Calculer $f'(\mathbf{x}; \mathbf{y})$ pour des vecteurs arbitraires $\mathbf{x}$ et $\mathbf{y}$.*
2. *Même question lorsque $f(\mathbf{x}) = \|\mathbf{x}\|^4$.*
3. *Prendre $n = 2$ pour cette dernière fonction et trouver l'ensemble des points tels que*

$$f'(2\mathbf{i} + 3\mathbf{j}; x\mathbf{i} + y\mathbf{j}) = 6.$$

**Exercice 3** *Dans chacun des exemples suivants, calculer les dérivées partielles du premier ordre des champs scalaires suivants*

1. $f(x, y) = x^2 + y^2 \sin(xy)$,
2. $f(x, y) = \dfrac{x + y}{x - y}$, *pour* $x \neq y$,
3. $f(\mathbf{x}) = \mathbf{x}_0 \cdot \mathbf{x}$, *le vecteur* $\mathbf{x}_0$ *étant fixé (forme linéaire),*
4. $f(\mathbf{x}) = \displaystyle\sum_{i,j=1}^{n} a_{i,j} x_i x_j$, *le vecteur* $\mathbf{x}_0$ *étant fixé (forme quadratique).*

### 1.3.4 Dérivées directionnelles et continuité

Dans le cas unidimensionnel, l'existence de la dérivée d'une fonction $f$ en un point implique la continuité en ce point. Ceci se montre facilement en prenant $h \neq 0$ et en écrivant

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

Lorsque $h \to 0$, le membre de droite tend vers $0$ $(= f'(x_0) \cdot 0)$ et ainsi $f(x_0 + h) \to_{h \to 0} f(x_0)$.

Appliquons maintenant le même argument à un champ scalaire général. Supposons que la dérivée $f'(\mathbf{x}_0; \mathbf{y})$ existe pour un certain $\mathbf{y}$. Alors, si $h \neq 0$, on peut écrire

$$f(\mathbf{x}_0 + h\mathbf{y}) - f(\mathbf{x}_0) = (\frac{f(\mathbf{x}_0 + h\mathbf{y}) - f(\mathbf{x}_0)}{h})h.$$

Lorsque $h$ tend vers 0, le membre de droite tend vers $f'(\mathbf{x}_0; \mathbf{y}) \cdot h = 0$ ; ainsi, l'existence de $f'(\mathbf{x}_0; \mathbf{y})$ pour un

$\mathbf{y}$ donné implique que

$$\lim_{h\to 0} f(\mathbf{x}_0 + h\mathbf{y}) = f(\mathbf{x}_0),$$

pour le même $\mathbf{y}$. Ceci signifie que $f(\mathbf{x}) \to f(\mathbf{x}_0)$ lorsque $\mathbf{x} \to \mathbf{x}_0$ le long d'une ligne droite passant par $\mathbf{x}_0$ et ayant comme direction $\mathbf{y}$. Si $f'(\mathbf{x}_0; \mathbf{y})$ existe pour tout vecteur $\mathbf{y}$, alors $f(\mathbf{x}) \to f(\mathbf{x}_0)$ lorsque $\mathbf{x} \to \mathbf{x}_0$ le long d'une ligne droite passant par $\mathbf{x}_0$. Ceci semble suggérer que $f$ est continue en $\mathbf{x}_0$. De manière assez surprenante, cette conclusion peut être fausse. Les exemples suivants décrivent des champs scalaires qui possèdent une dérivée directionnelle selon chaque direction partant de $\mathbf{0}$ mais qui ne sont pas continus en ce point.

Un premier exemple est donné par la fonction $f$ définie de la manière suivante

$$f(x, y) = 1 \text{ si } x < 0 \text{ ou } y > x^2,$$
$$f(x, y) = 0 \text{ sinon.}$$

Il est clair que ce champ admet une dérivée directionnelle (nulle) selon chaque direction. Toutefois, le champ n'est pas continu à l'origine.

Soit maintenant $f : \mathbb{R}^2 \to \mathbb{R}$ telle que

$$f(x, y) = \frac{xy^2}{x^2 + y^4}, \quad \text{si } x \neq 0,$$
$$f(0, y) = 0, \text{sinon.}$$

Soit $\mathbf{x}_0 = (0, 0)$ et $\mathbf{y} = (a, b)$ un vecteur. Si $a \neq 0$ et si $h \neq 0$, nous avons

$$\frac{f(\mathbf{0} + h\mathbf{y}) - f(\mathbf{0})}{h} = \frac{f(ha, hb)}{h} = \frac{ab^2}{a^2 + h^2 b^4}.$$

Soit $h \to 0$. Nous trouvons $f'(\mathbf{0}; \mathbf{y}) = \dfrac{b^2}{a}$. Si $\mathbf{y} = (0, b)$, nous trouvons, de manière similaire que $f'(\mathbf{0}; \mathbf{y}) = 0$. Ainsi, $f'(\mathbf{0}; \mathbf{y})$ existe pour toute direction $\mathbf{y}$. De plus, $f(\mathbf{x}) \to 0$ lorsque $\mathbf{x} \to 0$ le long de toute ligne droite partant de l'origine. Toutefois, en chaque point de la parabole $x = y^2$ (exepté à l'origine) la fonction a comme valeur $1/2$. Puisque de tels points existent arbitrairement proche de l'origine et que $f(\mathbf{0}) = 0$, la fonction $f$ n'est pas continue en $\mathbf{0}$. Cet exemple montre que l'existence de toutes les dérivées directionnelles n'implique pas la continuité en ce point. Pour cette raison, les dérivées directionnelles sont une extension insatisfaisante du concept de dérivée. Une généralisation plus satisfaisante existe. Elle implique la continuité et, simultanément, nous permet d'étendre les principaux résultats rencontrés dans le cas unidimensionnel à des dimensions supérieures. C'est ce que l'on appelle la dérivée totale.

### 1.3.5 La dérivée totale

Rappelons que, dans le cas unidimensionnel, une fonction $f$ qui admet une dérivée en un point $x_0$ peut être approchée près de ce point par une approximation linéaire de Taylor. Si $f'(x_0)$ existe, nous notons $E(x_0; h)$ la différence

$$E(x_0; h) = \frac{f(x_0 + h) - f(x_0)}{h} - f'(x_0), \quad \text{si } h \neq 0. \tag{1.4}$$

Définissons $E(x_0;0)=0$. De (1.4), nous obtenons la formule

$$f(x_0+h)=f(x_0)+f'(x_0)h+hE(x_0;h),$$

qui a également lieu pour $h=0$. C'est ce que l'on appelle le développement de Taylor du premier ordre pour approcher $f(x_0+h)-f(x_0)$ par $f'(x_0)h$. L'erreur commise est $hE(x_0;h)$. De (1.4), nous voyons que $E(x_0;h)\to 0$ lorsque $h\to 0$. En cela, l'erreur commise sur $hE(x_0;h)$ est d'ordre inférieur à $h$, pour $h$ petit. Ce point de vue qui consiste à approcher une fonction différentiable par une fonction linéaire suggère une manière d'étendre le concept de différentiabilité à des dimensions supérieures.

Soit $f:\mathcal{S}\to\mathbb{R}$ un champ scalaire défini sur un ensemble $\mathcal{S}$ de $\mathbb{R}^n$. Soit $\mathbf{x}_0$ un point intérieur de $\mathcal{S}$ et $\mathcal{B}(\mathbf{x}_0;r)$ une $n$-boule se trouvant dans $\mathcal{S}$. Soit $\mathbf{v}$ un vecteur tel que $\|\mathbf{v}\|<r$ ; ainsi, $\mathbf{x}_0+\mathbf{v}\in\mathcal{B}(\mathbf{x}_0;r)$.

**Définition 9** *Nous dirons que $f$ est différentiable en $\mathbf{x}_0$ si il existe une transformation linéaire*

$$T_{\mathbf{x}_0}:\mathbb{R}^n\to\mathbb{R},$$

*et une fonction scalaire $E(\mathbf{x}_0;\mathbf{v})$ telle que*

$$f(\mathbf{x}_0+\mathbf{v})=f(\mathbf{x}_0)+T_{\mathbf{x}_0}(\mathbf{v})+\|\mathbf{v}\|\,E(\mathbf{x}_0;\mathbf{v}),\tag{1.5}$$

*pour $\|\mathbf{v}\|<r$, où $\lim_{\|\mathbf{h}\|\to 0}E(\mathbf{x}_0;\mathbf{v})=0$. La transformation linéaire $T_{\mathbf{x}_0}$ est appelée la dérivée totale*[1] *de $f$ en $\mathbf{x}_0$.*

L'équation (1.5), qui a lieu pour $\|\mathbf{v}\|<r$, est appelée formule de Taylor du premier ordre pour $f(\mathbf{x}_0+\mathbf{v})$. Ceci donne une approximation linéaire, $T_{\mathbf{x}_0}(\mathbf{v})$, de la différence $f(\mathbf{x}_0+\mathbf{v})-f(\mathbf{x}_0)$. L'erreur dans l'approximation est $\|\mathbf{v}\|\,E(\mathbf{x}_0;\mathbf{v})$, un terme qui est d'ordre inférieur à $\|\mathbf{v}\|$ lorsque $\|\mathbf{v}\|\to 0$ (c'est à dire, $E(\mathbf{x}_0;\mathbf{v})=o(\|\mathbf{v}\|)$ lorsque $\|\mathbf{v}\|\to 0$. (Rappeler la notation) Le prochain théorème montre que, si la dérivée totale existe, alors elle est unique. Il nous dit également comment calculer la dérivée totale $T_{\mathbf{x}_0}(\mathbf{v})$, $\forall\mathbf{v}\in\mathbb{R}^n$.

**Théorème 4** *Supposons que $f$ soit différentiable en $\mathbf{x}_0$ et de dérivée totale $T_{\mathbf{x}_0}$. Alors, la dérivée $f'(\mathbf{x}_0;\mathbf{y})$ existe pour tout $\mathbf{y}\in\mathbb{R}^n$ et nous avons*

$$T_{\mathbf{x}_0}(\mathbf{y})=f'(\mathbf{x}_0;\mathbf{y}).$$

*De plus, $f'(\mathbf{x}_0;\mathbf{y})$ est une combinaison linéaire des composantes de $\mathbf{y}$. En fait, en posant $\mathbf{y}=(y_1,...,y_n)$, nous avons*

$$f'(\mathbf{x}_0;\mathbf{y})=\sum_{k=1}^{n}D_kf(\mathbf{x}_0)y_k=Df(\mathbf{x}_0)\cdot\mathbf{y}.\tag{1.6}$$

### 1.3.6 Le gradient d'un champ scalaire

Nous pouvons récrire (1.6) comme

$$f'(\mathbf{x}_0;\mathbf{y})=\nabla f(\mathbf{x}_0)\cdot\mathbf{y},$$

[1] La dérivée totale n'est pas un nombre mais une application linéaire. La quantité $T_{\mathbf{x}_0}(\mathbf{v})$ est un nombre ; il est défini pour tout point $\mathbf{v}$ de $\mathbb{R}^n$. La dérivée totale a été introduite par Young en 1908 puis par Fréchet en 1911.

où $\nabla f(\mathbf{x}_0)$ est le vecteur dont les composantes sont les dérivées partielles de $f$ en $\mathbf{x}_0$

$$\nabla f(\mathbf{x}_0) = (D_1 f(\mathbf{x}_0), ..., D_n f(\mathbf{x}_0)).$$

C'est ce que l'on appelle le gradient de $f$. Le gradient $\nabla f$ est un champ scalaire défini en chaque point $\mathbf{x}_0$ où les dérivées partielles $D_1 f(\mathbf{x}_0), ..., D_n f(\mathbf{x}_0)$ existent. On trouve également quelques fois la notation grad$f$. Nous pouvons, sous ces notations, récrire la formule de Taylor au premier ordre

$$f(\mathbf{x}_0 + \mathbf{v}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot \mathbf{v} + \|\mathbf{v}\| \, E(\mathbf{x}_0; \mathbf{v}), \tag{1.7}$$

où $E(\mathbf{x}_0; \mathbf{v}) \rightarrow_{\|\mathbf{v}\| \rightarrow 0} 0$. Cette forme est alors similaire à celle obtenue dans le cas unidimensionnel où $\nabla f(\mathbf{x}_0)$ joue le rôle de $f'(\mathbf{x}_0)$. A partir de la formule de Taylor, on peut montrer que la différentiabilité implique la continuité.

**Théorème 5** *Si un champ $f$ est différentiable en $\mathbf{x}_0$, alors $f$ est continue en $\mathbf{x}_0$.*

Dans le cas bidimensionnel, le vecteur gradient s'écrit

$$\nabla f(x, y) = \frac{\partial f}{\partial x}(x, y)\mathbf{i} + \frac{\partial f}{\partial y}(x, y)\mathbf{j}.$$

En dimension trois, nous avons

$$\nabla f(x, y, z) = \frac{\partial f}{\partial x}(x, y, z)\mathbf{i} + \frac{\partial f}{\partial y}(x, y, z)\mathbf{j} + \frac{\partial f}{\partial z}(x, y, z)\mathbf{k}.$$

### 1.3.7 Une condition suffisante de différentiabilité

Si $f$ est différentiable en $\mathbf{x}_0$, alors toutes les dérivées partielles $D_1 f(\mathbf{x}_0)$,...,$D_n f(\mathbf{x}_0)$ existent. Toutefois, l'inverse est faux (cf. le contre-exemple précédent).

Le prochain théorème montre que l'existence de dérivées partielles continues en un point implique la différentiabilité en ce point.

**Théorème 6** *(Une condition suffisante de différentiabilité.) Supposons que les dérivées partielles $D_1 f(\mathbf{x}_0)$,...,$D_n f(\mathbf{x}_0)$ existent dans une $n$-boule $\mathcal{B}(\mathbf{x}_0)$ et sont continues en $\mathbf{x}_0$. Alors, $f$ est différentiable en $\mathbf{x}_0$.*

*Preuve.* Soit $y \in \mathcal{B}$ fixé, on considère $g : [0, 1] \rightarrow \mathbb{R}$ définie par $g(h) = f(x_0 + h.y)$. On vérifie que $g \in \mathcal{C}^1[0, 1]$. En utilisant la formule des accroissements finis pour $g$ entre 0 et 1, nous obtenons bien l'existence de $\theta = \theta(y) \in ]0, 1[$ tel que :

$$f(x_0 + y) - f(x_0) = \nabla f(x_0 + \theta y).y$$

Ainsi

$$w(x_0, y) := f(x_0 + y) - f(x_0) - \nabla f(x_0).y = (\nabla f(x_0 + \theta y) - \nabla f(x_0)).y$$

Les hypothèses faites sur $f$ impliquent donc que $w(x_0, y)/\|y\|$ tend vers 0 lorsque $y$ tend vers 0. Ceci montre encore que $f$ est différentiable en $x_0$ et que la différentielle (i.e. la dérivée totale) est donnée par $Df(x_0)y = \nabla f(x_0).y$. ■

**Remarque 1** *Un champ scalaire satisfaisant les hypothèses du théorème précédent est appelé continuement différentiable en* $\mathbf{x}_0$.

**Exercice 4** *Donner les gradients en chacun des points où il existe pour les fonctions suivantes*

1. $f(x, y) = x^2 + y^2 \sin(xy)$,
2. $f(x, y, z) = x^2 - y^2 + 2z^2$.

**Exercice 5** *Evaluer la dérivée directionnelle du champ scalaire* $f(x, y, z) = x^2 + y^2 + 3z^2$ *au point* $(1, 1, 0)$ *selon la direction* $\mathbf{y} = \mathbf{i} - \mathbf{j} + 2\mathbf{k}$.

**Exercice 6** *Dans* $\mathbb{R}^3$, *soit* $\mathbf{r}(x, y, z) = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, *et* $r = \|\mathbf{r}\|$.

1. *Montrer que* $\nabla r$ *est un vecteur unitaire dans la direction de* $\mathbf{r}$.
2. *Montrer que* $\nabla r^n = nr^{n-2}\mathbf{r}$, *avec* $n$ *un entier strictement positif.*
3. *Montrer que la formule reste vraie pour* $n \leq 0$.
4. *trouver un champ scalaire* $f$ *tel que* $\nabla f = \mathbf{r}$.

## 1.4 Quelques règles et résultats utiles

### 1.4.1 Une règle de dérivation en chaîne pour les champs scalaires

Dans le cas unidimensionnel, la règle de calcul en chaîne permet de calculer la dérivée d'une fonction composée $g(t) = f(r(t))$ par la formule

$$g'(t) = f'(r(t))r'(t).$$

Nous allons étendre cette formule lorsque

- $f$ est remplacée par un champ scalaire défini sur un ensemble d'un espace $n$-dimensionnel,
- $r$ est remplacée par une fonction d'une variable réelle à valeurs dans le domaine de $f$.

Nous verrons ensuite comment étendre la formule pour $f$ et $r$ des champs vectoriels.

**Théorème 7** *Soit* $f$ *un champ scalaire défini sur un ensemble ouvert* $\mathcal{S}$ *dans* $\mathbb{R}^n$ *et soit* $\mathbf{r}$ *une fonction à valeurs vectorielles réelles qui transporte un intervalle* $J$ *de* $\mathbb{R}$ *dans* $\mathcal{S}$. *Définissons la fonction composée* $g = f \circ \mathbf{r}$ *sur* $J$ *par la relation*

$$g(t) = f(\mathbf{r}(t)), \quad si\ t \in J.$$

*Soit* $t_0$ *un point de* $J$ *où* $\mathbf{r}'(t_0)$ *existe et supposons que* $f$ *est différentiable en* $\mathbf{r}(t_0)$. *Alors,* $g'(t_0)$ *existe et est égale au produit scalaire*

$$g'(t_0) = \nabla f(\mathbf{x}_0) \cdot \mathbf{r}'(t_0), \quad où\ \mathbf{x}_0 = \mathbf{r}(t_0).$$

**Exercice 7** *On suppose que toutes les dérivées des fonctions suivantes sont continues et existent. Les équations* $u = f(x, y)$, $x = X(t)$, $y = Y(t)$, *définissent* $u$ *comme une fonction de* $t$, *que nous notons* $u = F(t)$.

*a) Utiliser la règle de dérivation en chaîne pour montrer que*

$$F'(t) = \frac{\partial f}{\partial x} X'(t) + \frac{\partial f}{\partial y} Y'(t),$$

*où* $\dfrac{\partial f}{\partial x}$ *et* $\dfrac{\partial f}{\partial y}$ *sont évaluées en* $(X(t), Y(t))$.

*b) De la même manière, calculer* $F''(t)$ *en fonction de* $f$, $X$ *et* $Y$.

*c) Appliquer ces résultats aux fonctions*

$$f(x, y) = x^2 + y^2, \quad X(t) = t, \quad Y(t) = t^2.$$

### 1.4.2 Dérivée d'un champ vectoriel

La théorie de la dérivation pour les champs de vecteurs est une extension directe de celle pour les champs scalaires. Soit $\mathbf{f} : \mathcal{S} \to \mathbb{R}^m$ un champ de vecteurs défini sur un sous-ensemble $\mathcal{S}$ de $\mathbb{R}^n$. Si $\mathbf{x}_0$ est un point intérieur de $\mathcal{S}$ et si $\mathbf{y}$ est un vecteur de $\mathbb{R}^n$, on définit la dérivée totale $\mathbf{f}'(\mathbf{x}_0; \mathbf{y})$ par la formule

$$\mathbf{f}'(\mathbf{x}_0; \mathbf{y}) = \lim_{h \to 0} \frac{\mathbf{f}(\mathbf{x}_0 + h\mathbf{y}) - \mathbf{f}(\mathbf{x}_0)}{h}$$

dès que la limite existe. La dérivée $\mathbf{f}'(\mathbf{x}_0; \mathbf{y})$ est un vecteur de $\mathbb{R}^m$.

Soit $f_k$ la $k$-ième composante de $\mathbf{f}$. Remarquons que la dérivée $\mathbf{f}'(\mathbf{x}_0; \mathbf{y})$ existe si et seulement si $f_k'(\mathbf{x}_0; \mathbf{y})$ existe pour chaque $1 \leq k \leq m$, auquel cas nous avons

$$\mathbf{f}'(\mathbf{x}_0; \mathbf{y}) = (f_1'(\mathbf{x}_0; \mathbf{y}), ..., f_m'(\mathbf{x}_0; \mathbf{y})) = \sum_{k=1}^{m} f_k'(\mathbf{x}_0; \mathbf{y}) \mathbf{e}_k,$$

où $\mathbf{e}_k$ est le $k$-ième vecteur des coordonnées.

Nous dirons que $\mathbf{f}$ est différentiable en un point $\mathbf{x}_0$ si il existe une transformation linéaire

$$T_{\mathbf{x}_0} : \mathbb{R}^n \to \mathbb{R}^m$$

telle que

$$\mathbf{f}(\mathbf{x}_0 + \mathbf{v}) = \mathbf{f}(\mathbf{x}_0) + T_{\mathbf{x}_0}(\mathbf{v}) + \|\mathbf{v}\| \, \mathbf{E}(\mathbf{x}_0; \mathbf{v}), \tag{1.8}$$

où $\mathbf{E}(\mathbf{x}_0; \mathbf{v}) \to 0$ lorsque $\mathbf{v} \to \mathbf{0}$. La formule de Taylor au premier ordre (1.8) a lieu pour tout $\mathbf{v}$ tel que $\|\mathbf{v}\| < r$ pour des $r > 0$. Le terme $\mathbf{E}(\mathbf{x}_0; \mathbf{v})$ est un vecteur de $\mathbb{R}^m$. La transformation linéaire $T_{\mathbf{x}_0}$ est appelée dérivée totale de $\mathbf{f}$ en $\mathbf{x}_0$. Pour des champs scalaires, nous avons prouvé que $T_{\mathbf{x}_0}(\mathbf{y})$ est le produit scalaire du gradient $\nabla f(\mathbf{x}_0)$ avec $\mathbf{y}$. Pour les champs de vecteurs, nous allons prouver que $T_{\mathbf{x}_0}$ est un vecteur dont la $k$-ième composante est le produit scalaire $\nabla f_k(\mathbf{x}_0) \cdot \mathbf{y}$.

**Théorème 8** *Supposons que* $\mathbf{f}$ *soit différentiable en* $\mathbf{x}_0$ *et de dérivée totale* $T_{\mathbf{x}_0}$. *Alors, la dérivée* $\mathbf{f}'(\mathbf{x}_0; \mathbf{y})$

*existe pour tout* $\mathbf{x}_0$ *dans* $\mathbb{R}^n$ *et nous avons*

$$T_{\mathbf{x}_0}(\mathbf{y}) = \mathbf{f}'(\mathbf{x}_0; \mathbf{y}).$$

*De plus, si* $\mathbf{f} = (f_1, ..., f_m)$ *et si* $\mathbf{y} = (y_1, ..., y_n)$*, nous avons*

$$T_{\mathbf{x}_0}(\mathbf{y}) = \sum_{k=1}^{m} \nabla f_k(\mathbf{x}_0) \cdot \mathbf{y}\mathbf{e}_k = (\nabla f_1(\mathbf{x}_0) \cdot \mathbf{y}, ..., \nabla f_m(\mathbf{x}_0) \cdot \mathbf{y}). \tag{1.9}$$

On peut en fait récrire plus simplement sous forme matricielle cette dernière relation

$$T_{\mathbf{x}_0}(\mathbf{y}) = Df(\mathbf{x}_0)\mathbf{y},$$

où $Df(\mathbf{x}_0)$ est la matrice $m \times n$ dont la $k$-ième ligne est $\nabla f_k(\mathbf{x}_0)$ et où $\mathbf{y}$ est regardé comme un vecteur de longueur $n$. La matrice $Df(\mathbf{x}_0)$ est appelée matrice jacobienne de $\mathbf{f}$ en $\mathbf{x}_0$. Son $j$-ème élément est la dérivée partielle $D_j f_k(\mathbf{x}_0)$. Ainsi, nous avons

$$Df(\mathbf{x}_0) = \begin{pmatrix} D_1 f_1(\mathbf{x}_0) ... D_n f_1(\mathbf{x}_0) \\ ..... \\ D_1 f_m(\mathbf{x}_0) ... D_n f_m(\mathbf{x}_0) \end{pmatrix}$$

La matrice jacobienne $Df(\mathbf{x}_0)$ est définie en chacun des points où les $mn$ dérivées partielles $D_j f_k(\mathbf{x}_0)$ existent. La dérivée totale $T_{\mathbf{x}_0}$ s'écrit également $\mathbf{f}'(\mathbf{x}_0)$. La dérivée $\mathbf{f}'(\mathbf{x}_0)$ est une transformation linéaire ; le jacobien $Df(\mathbf{x}_0)$ est une représentation matricielle de cette transformation. La formule de Taylor au premier ordre s'écrit

$$\mathbf{f}(\mathbf{x}_0 + \mathbf{v}) = \mathbf{f}(\mathbf{x}_0) + \mathbf{f}'(\mathbf{x}_0)(\mathbf{v}) + \|\mathbf{v}\| \, \mathbf{E}(\mathbf{x}_0; \mathbf{v}),$$

où $\mathbf{E}(\mathbf{x}_0; \mathbf{v}) \to \mathbf{0}$ lorsque $\mathbf{v} \to \mathbf{0}$. Pour calculer les composantes du vecteur $\mathbf{f}'(\mathbf{x}_0)(\mathbf{v})$, on peut utiliser le produit matriciel $Df(\mathbf{x}_0)\mathbf{v}$ ou encore la formule (1.9).

De manière similaire au cas scalaire, nous avons que la différentiabilité d'un champ vectoriel implique la continuité de ce champ.

**Théorème 9** *Si un champ de vecteurs* $\mathbf{f}$ *est différentiable en* $\mathbf{x}_0$*, alors* $\mathbf{f}$ *est continue en* $\mathbf{x}_0$*.*

### 1.4.3 La règle de dérivation en chaîne pour les champs de vecteurs

**Théorème 10** *(dérivation en chaîne). Soient* $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ *et* $\mathbf{g} : \mathbb{R}^p \to \mathbb{R}^n$ *des champs de vecteurs tels que la composition* $\mathbf{h} = \mathbf{f} \circ \mathbf{g}$ *soit définie dans un voisinage d'un point* $\mathbf{x}_0$*. Supposons que* $\mathbf{g}$ *soit différentiable dans un voisinage d'un point* $\mathbf{x}_0$*, de dérivée totale* $\mathbf{g}'(\mathbf{x}_0)$*. Soit* $\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0)$ *et supposons que* $\mathbf{f}$ *soit différentiable en* $\mathbf{y}_0$*, de dérivée totale* $\mathbf{f}'(\mathbf{y}_0)$*. Alors,* $\mathbf{h}$ *est différentiable en* $\mathbf{x}_0$*, et la dérivée totale* $\mathbf{h}'(\mathbf{x}_0)$ *est donnée par*

$$\mathbf{h}'(\mathbf{x}_0) = \mathbf{f}'(\mathbf{y}_0) \circ \mathbf{g}'(\mathbf{x}_0).$$

Soit $\mathbf{h} = \mathbf{f} \circ \mathbf{g}$, où $\mathbf{g}$ est différentiable en $\mathbf{x}_0$ et $\mathbf{f}$ est différentiable en $\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0)$. La dérivation en chaîne nous donne

$$\mathbf{h}'(\mathbf{x}_0) = \mathbf{f}'(\mathbf{y}_0) \circ \mathbf{g}'(\mathbf{x}_0).$$

On peut en fait exprimer la règle de dérivation en chaîne grâce aux matrices jacobiennes $Dh(\mathbf{x}_0)$, $Df(\mathbf{y}_0)$ et $Dg(\mathbf{x}_0)$ qui représentent les transformations linéaires $\mathbf{h}'(\mathbf{x}_0)$, $\mathbf{f}'(\mathbf{y}_0)$ et $\mathbf{g}'(\mathbf{x}_0)$, respectivement. Puisque la composition d'applications correspond à la multiplication de matrices, nous obtenons

$$Dh(\mathbf{x}_0) = Df(\mathbf{y}_0)Dg(\mathbf{x}_0), \tag{1.10}$$

où $\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0)$. On peut également exprimer la règle sous la forme d'un ensemble d'équations scalaires. Supposons que $\mathbf{x}_0 \in \mathbb{R}^p$, $\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0) \in \mathbb{R}^n$, et $\mathbf{f}(\mathbf{y}_0) \in \mathbb{R}^m$. Alors, $\mathbf{h}(\mathbf{x}_0) \in \mathbb{R}^m$ et nous pouvons écrire

$$\begin{aligned}\mathbf{g} &= (g_1, ..., g_n),\\ \mathbf{f} &= (f_1, ..., f_m),\\ \mathbf{g} &= (h_1, ..., h_m).\end{aligned}$$

Alors, $Dh(\mathbf{x}_0)$ est une matrice $m \times p$, $Dh(\mathbf{y}_0)$ est une matrice $m \times n$ et $Dg(\mathbf{x}_0)$ est une matrice $n \times p$, données respectivement par

$$\begin{aligned}Dh(\mathbf{x}_0) &= [D_j h_i(\mathbf{x}_0)]_{i,j=1}^{m,p},\\ Df(\mathbf{x}_0) &= [D_j f_i(\mathbf{y}_0)]_{i,j=1}^{m,n},\\ Dg(\mathbf{x}_0) &= [D_j g_i(\mathbf{x}_0)]_{i,j=1}^{n,p}.\end{aligned}$$

L'équation (1.10) est équivalente à $mp$ équations scalaires

$$D_j h_i(\mathbf{x}_0) = \sum_{k=1}^{n} D_k f_i(\mathbf{y}_0) D_j g_k(\mathbf{x}_0), \quad \forall 1 \leq i \leq m, \quad 1 \leq j \leq p.$$

**Exemple 5** *(Règle de dérivation en chaîne développée pour les champs scalaires). Si* $m = 1$, $\mathbf{f}$ *est un champ scalaire,* $\mathbf{h}$ *aussi. Les* $p$ *équations (une pour chaque dérivée partielle de* $\mathbf{h}$*) s'écrivent*

$$D_j h(\mathbf{x}_0) = \sum_{k=1}^{n} D_k f(\mathbf{y}_0) D_j g_k(\mathbf{x}_0), \quad \forall 1 \leq j \leq p.$$

*Le cas particulier* $p = 1$ *donne une équation*

$$h'(\mathbf{x}_0) = \sum_{k=1}^{n} D_k f(\mathbf{y}_0) g_k'(\mathbf{x}_0).$$

**Exemple 6** *Considérons* $p = 2$ *et* $n = 2$*. Ecrivons* $\mathbf{x}_0 = (s, t)$ *et* $\mathbf{b} = (x, y)$*. Alors, les composantes de* $x$ *et* $y$ *sont reliées à* $s$ *et* $t$ *par les équations*

$$x = g_1(s, t) \quad et \quad y = g_2(s, t).$$

*La règle de dérivation en chaîne donne un couple d'équations pour les dérivées partielles de* $\mathbf{h}$

$$D_1\mathbf{h}(s,t) = D_1(\mathbf{f}\circ\mathbf{g}) = D_1\mathbf{f}(x,y)D_1g_1(s,t) + D_2\mathbf{f}(x,y)D_1g_2(s,t).$$

*De même, nous avons*

$$D_2\mathbf{h}(s,t) = D_1\mathbf{f}(x,y)D_2g_1(s,t) + D_2\mathbf{f}(x,y)D_2g_2(s,t).$$

*Selon la notation* $\partial$*, ce couple d'équations s'écrit*

$$\begin{aligned}\frac{\partial f}{\partial s} &= \frac{\partial f}{\partial x}\frac{\partial g}{\partial s} + \frac{\partial f}{\partial y}\frac{\partial g}{\partial s},\\ \frac{\partial f}{\partial t} &= \frac{\partial f}{\partial x}\frac{\partial g}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial g}{\partial t}.\end{aligned}$$

**Exercice 8** *(Coordonnées polaires). Soit* $f$ *un champ scalaire dépendant de* $(x,y)$*. En coordonnées polaires, nous avons* $x = r\cos(\theta)$ *et* $y = r\sin(\theta)$*. Posons :* $\varphi(r,\theta) = f(r\cos(\theta), r\sin(\theta))$*.*

- *Exprimer* $\dfrac{\partial\varphi}{\partial r}$ *et* $\dfrac{\partial\varphi}{\partial\theta}$ *en fonction de* $\dfrac{\partial f}{\partial x}$ *et* $\dfrac{\partial f}{\partial y}$*.*
- *Exprimer la dérivée partielle du second ordre* $\dfrac{\partial^2\varphi}{\partial\theta^2}$ *en fonction de celles de* $f$*.*

### 1.4.4 Conditions suffisantes pour avoir l'égalité des dérivées partielles mixtes

Si $f$ est une fonction à valeurs réelles de deux variables, les deux dérivées mixtes $D_{1,2}f$ et $D_{2,1}f$ ne sont pas nécessairement égales. Un exemple est donné par l'exercice suivant.

**Exercice 9** *Soit* $f$ *la fonction définie par*

$$f(x,y) = xy\frac{x^2-y^2}{x^2+y^2}, \quad \textit{pour } (x,y)\neq(0,0),$$

*et* $f(0,0) = 0$ *sinon. Montrer que* $D_{2,1}f(0,0) = -1$ *et* $D_{1,2}f(0,0) = 1$*.*

Dans l'exercice précédent, les deux dérivées partielles $D_{2,1}f$ et $D_{1,2}f$ ne sont pas continues à l'origine. On peut montrer que les deux dérivées mixtes sont égales en un point $(x_0,y_0)$ si au moins l'une d'entre elles est continue en un voisinage de ce point. On peut montrer dans un premier temps qu'elles sont égales si elles sont toutes deux continues. Ceci fait l'objet du théorème suivant.

**Théorème 11** *(Une condition suffisante pour l'égalité des dérivées partielles mixtes). Supposons que* $f$ *soit un champ scalaire tel que les dérivées partielles* $D_1f$*,* $D_2f$*,* $D_{1,2}f$ *et* $D_{2,1}f$ *existent sur un ouvert* $\mathcal{S}$*. Si* $(x_0,y_0)$ *est un point de* $\mathcal{S}$ *où* $D_{1,2}f$ *et* $D_{2,1}f$ *sont continues, nous avons alors*

$$D_{1,2}f(x_0,y_0) = D_{2,1}f(x_0,y_0).$$

On peut en fait démontrer une version plus forte de ce théorème.

**Théorème 12** *Soit $f$ un champ scalaire tel que les dérivées partielles $D_1 f$, $D_2 f$ et $D_{2,1} f$ existent sur un ouvert $\mathcal{S}$ contenant $(x_0, y_0)$. Supposons de plus que $D_{2,1} f$ est continue sur $\mathcal{S}$. Alors, les dérivées partielles $D_{1,2} f(x_0, y_0)$ existent et nous avons*

$$D_{1,2} f(x_0, y_0) = D_{2,1} f(x_0, y_0).$$

## 1.5 Exercices

**Exercice 1.1** (Calcul explicite de différentielles).
Calculer la différentielle à l'origine des applications suivantes :

$$\begin{array}{rccl}
(i)\ f: & \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\
& (x,y) & \longmapsto & f(x,y) = x^3 + \sqrt{1+x^2+y^2} \\
(ii)\ g: & \mathbb{R}^3 & \longrightarrow & \mathbb{R} \\
& (x,y,z) & \longmapsto & g(x,y,z) = xyz\sin(xy) + 2x + 5.
\end{array}$$

**Exercice 1.2** (Linéarité des opérateurs différentiels).
On définit les fonctions $\triangle$ et $\Phi$ par :

$$\begin{array}{rccl}
\triangle: & \mathcal{C}^\infty(\mathbb{R}^2) & \longrightarrow & \mathcal{C}^\infty(\mathbb{R}^2) \\
& \varphi & \longmapsto & \dfrac{\partial^2 \varphi}{\partial x^2} + \dfrac{\partial^2 \varphi}{\partial y^2}.
\end{array}
\qquad \text{et } \Phi: \quad
\begin{array}{rccl}
& \mathcal{C}^\infty(\mathbb{R}^2) & \longrightarrow & \mathcal{C}^\infty(\mathbb{R}^2) \\
& \varphi & \longmapsto & \dfrac{\partial \varphi}{\partial x} \cdot \dfrac{\partial \varphi}{\partial y}.
\end{array}$$

$\triangle$ et $\Phi$ sont-elles linéaires ?

**Exercice 1.3** (Continuité et différentiabilité d'une fonction de plusieurs variables).
Soit $f$, la fonction définie par :

$$\begin{array}{rccl}
f: & \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\
& (x,y) & \longmapsto & \left\{ \begin{array}{ll} 0 & \text{si} \quad (x,y) = (0,0) \\ \dfrac{|x|^{\frac{3}{2}} y}{x^2+y^2} & \text{si} \quad (x,y) \neq (0,0) \end{array} \right.
\end{array}$$

1. $f$ est-elle continue ?
2. Calculer les dérivées partielles de $f$. Sont-elles continues ?
3. Calculer les dérivées directionnelles (si elles existent) de $f$ au point de coordonnées $(0,0)$. L'application $f$ est-elle différentiable en $(0,0)$ ?

**Exercice 1.4** (Continuité et différentiabilité d'une fonction de plusieurs variables).

On définit l'application $f$ de la façon suivante :

$$\begin{array}{rccc} f: & \mathbb{R}^2\backslash\{(0,0)\} & \longrightarrow & \mathbb{R} \\ & (x,y) & \longmapsto & \dfrac{xy}{\sqrt{x^2+y^2}}. \end{array}$$

1. Montrer que l'on peut prolonger $f$ par continuité. On appelle $\widetilde{f}$, ce prolongement.
2. Étudier la différentiabilité de $\widetilde{f}$.
3. $\widetilde{f}$ admet-elle des dérivées partielles ?
4. $f$ est-elle $\mathcal{C}^1$ sur son ensemble de définition ?

**Exercice 1.5** (Continuité et différentiabilité d'une fonction de plusieurs variables).
Reprendre les questions de l'exercice précédent avec la fonction :

$$\begin{array}{rccc} g: & \mathbb{R}^2\backslash\{(0,0)\} & \longrightarrow & \mathbb{R} \\ & (x,y) & \longmapsto & (x^4+y^4)\sin\left(\dfrac{1}{\sqrt{x^4+y^4}}\right). \end{array}$$

**Exercice 1.6** (Différentiabilité d'une fonction de plusieurs variables).
La fonction suivante est-elle différentiable ?

$$\begin{array}{rccc} f: & \mathbb{R}^2\backslash\{(0,0)\} & \longrightarrow & \mathbb{R} \\ & (x,y) & \longmapsto & f(x,y) = \left\{ \begin{array}{ll} \dfrac{y^2-x}{x^2+y^2} & \text{si} \quad (x,y)\neq(0,0) \\ 0 & \text{sinon} \end{array} \right. \end{array}$$

**Exercice 1.7** (Régularité d'une fonction de plusieurs variables).
Soit $f$, la fonction de $\mathbb{R}^2$ dans $\mathbb{R}$ définie par :

$$f(x,y) = x^2 y \sin\left(\frac{y}{x}\right).$$

1. Montrer que l'on peut définir un prolongement par continuité de la fonction $f$.
2. $f$ admet-elle des dérivées partielles sur $\mathbb{R}^2$ ?
3. $f$ est-elle de classe $\mathcal{C}^1$ sur $\mathbb{R}^2$ ?
4. Calculer $\dfrac{\partial^2 f}{\partial x \partial y}(0,0)$ et $\dfrac{\partial^2 f}{\partial y \partial x}(0,0)$.
   **Remarque :** attention à donner un sens convenable aux deux expressions ci-dessus, avant de les calculer. Rien ne certifie que $f$ est de classe $\mathcal{C}^2$ au voisinage de $(0,0)$.

**Exercice 1.8** (Étude complète d'une fonction de plusieurs variables).

Le but de ce problème est d'étudier, suivant les valeurs de $\alpha > 0$, la différentiabilité au point $(0,0)$ de la fonction de deux variables définie par :

$$f(x,y) = \frac{|x|^\alpha |y|^\alpha}{x^2 + y^2 - xy}.$$

On rappelle que le nombre $a^\alpha$ est défini pour tout $\alpha \in \mathbb{R}$ et $a > 0$ par la relation : $a^\alpha = e^{\alpha \ln a}$.

1. Déterminer l'ensemble de définition de $f$, noté $\mathcal{D}_f$.
2. Démontrer **rapidement** l'inégalité vérifiée pour tous $(x,y) \in \mathbb{R}^2$ : $|xy| \leq \frac{1}{2}\phi^2(x,y)$. En déduire un encadrement pour tous $(x,y)$ non nuls du nombre $\dfrac{f(x,y)}{|x|^\alpha |y|^\alpha}$.
3. **Étude de la continuité de $f$ en** $(0,0)$
   (a) **Cas $\alpha > 1$.**
   En utilisant la question précédente, démontrer que pour tous $(x,y)$ non nuls, on a :
   $$|f(x,y)| \leq \frac{1}{2^{\alpha-1}}\phi^{2(\alpha-1)}(x,y).$$
   Conclure.
   (b) **Cas $0 < \alpha \leq 1$.**
   Utiliser un arc paramétré pour prouver que $f$ est discontinue en 0 si $\alpha \in ]0;1]$.
   (c) Conclure.
4. **Étude de la différentiabilité de $f$ en** $(0,0)$
   (a) Donner un encadrement, pour $(x,y) \in \mathbb{R}^2 \backslash \{(0,0)\}$ du nombre : $\left| \dfrac{f(x,y)}{\phi(x,y)} \right|$.
   (b) Prouver que si $\alpha > \frac{3}{2}$, la fonction $f$ est différentiable en $(0,0)$.
   (c) En utilisant des arcs paramétrés, démontrer que la fonction $f$ est différentiable en $(0,0)$ si, et seulement si $\alpha > \frac{3}{2}$.
5. **Application : $\alpha = \frac{5}{4}$.** $f$ est-elle de classe $\mathcal{C}^1$ sur $\mathbb{R}^2$ ?

**Exercice 1.9** (Différentiabilité d'une fonction définie à l'aide d'un max). On définit la fonction $f$ sur $\mathbb{R}^2$ par : $f(x,y) = \max(x,y)$.

1. Par un système de coloriage **dans le plan**, trouver un moyen de représenter $f(x,y)$ en fonction de $x$ et $y$.
2. Démontrer que : $\forall (x,y) \in \mathbb{R}^2$, $f(x,y) = \dfrac{x + y + |x - y|}{2}$.
3. Étudier la différentiabilité de $f$ sur $\mathbb{R}^2$.

**Exercice 1.10** Liens entre les notions de dérivabilités

Soit $x_0 \in \mathbb{R}^n$ et $\mathcal{V} \subset \mathbb{R}^n$ un voisinnage ouvert de $x_0$. On considère $f : \mathcal{V} \subset \mathbb{R}^n \to \mathbb{R}$ et on introduit les

propriétés ci-dessous :

$(p_0)$ : $f$ est continue en $x_0$
$(p_1)$ : $f$ est différentiable en $x_0$
$(p_2)$ : $f$ est Gâteaux-dérivable en $x_0$
$(p_3)$ : $f$ est dérivable en $x_0$ par rapport à toutes les directions $y \in \mathbb{R}^n$.
$(p_4)$ : $f$ admet n dérivées partielles $\frac{\partial f}{\partial x_1}, .., \frac{\partial f}{\partial x_n}$ en $x_0$.
$(p_5)$ : $f$ admet n dérivées partielles en $x_0$ qui sont continues sur un voisinnage $\mathcal{V}'$ de $x_0$.

On demande de montrer les relations énoncées ci-dessous. Pour prouver les non équivalences on pourra donner des contre exemples.

1. Liens entre dérivabilité et continuité
   (a) $(p_1) \Rightarrow (p_0)$
   (b) $(p_2) \not\Rightarrow (p_0)$
2. Liens entre les différentes notions de dérivabilités
   (a) $(p_1) \Rightarrow (p_2) \Rightarrow (p_3) \Rightarrow (p_4)$ et $Df(x_0)y = f'(x_0; y) = \nabla f(x_0).y$
   (b) $(p_2) \not\Rightarrow (p_1)$
   (c) $(p_3) \not\Rightarrow (p_2)$
   (d) $(p_4) \not\Rightarrow (p_3)$
   (e) $(p_5) \Rightarrow (p_1)$

# Chapitre 2

# Compléments en calcul différentiel

Cette partie vient en complément de la partie calcul différentiel proposée en OCI (Outils de Calculs pour l'Ingénieur) et enseignée l'ENSEM en début de premier semestre.

## 2.1 Courbes de niveau

Nous avons vu, précédemment, la règle de dérivation en chaîne pour les champs scalaires. Cette règle peut être utilisée afin de déduire des propriétés du champ de gradient associé. Considérons $f$, un champ scalaire de $S$ dans $\mathbb{R}^n$, et définissons les points $\mathbf{x}$ de $S$ tels que $f(\mathbf{x})$ soit constant, c'est-à-dire, $f(\mathbf{x}) = c$. Soit $\mathcal{L}(c)$ cet ensemble $\mathcal{L}(c) := \{\mathbf{x} \in S \ \ / f(\mathbf{x}) = c\}$. L'ensemble $\mathcal{L}(c)$ est communément appelé ensemble de niveau de $f$. Dans $\mathbb{R}^2$, $\mathcal{L}(c)$ est appelé courbe de niveau et surface de niveau dans l'espace tridimensionnel.

Les lignes de niveau interviennent dans plusieurs domaines de la physique. Soit par exemple $f(x, y)$ la température en un point $(x, y)$. Les courbes de niveau de $f$ sont alors appelées isothermes. On sait que le flot de chaleur est dirigé selon la direction où la variation de température est la plus importante. Cette direction est connue pour être normale aux isothermes, elle décrit les lignes de flot, trajectoires orthogonales aux isothermes.

Considérons dorénavant un champ scalaire $f$ différentiable sur un ouvert $S$ de $\mathbb{R}^3$ et analysons en détails les surfaces de niveau $\mathcal{L}(c)$. Soit $\mathbf{x}_0$ un point de cette surface et $\Gamma$ une courbe se trouvant sur $S$ qui passe par $\mathbf{x}_0$. Alors, $\nabla f(\mathbf{x}_0)$ est normal à cette courbe en $\mathbf{x}_0$, ou encore, $\nabla f(\mathbf{x}_0)$ est perpendiculaire au plant tangent $T_{\mathbf{x}_0}$ de $\Gamma$ en $\mathbf{x}_0$. En effet, si nous supposons $\Gamma$ paramétrisée par un champ différentiable $\boldsymbol{\gamma}$ défini sur $I \subseteq \mathbb{R}$, puisque $\Gamma$ se trouve sur la surface de niveau $\mathcal{L}(c)$, nous avons : $g(t) = f(\boldsymbol{\gamma}(t)) = c$, pour tout $t \in I$. La règle de dérivation en chaîne (cf. le théorème 7) nous donne alors : $g'(t) = \nabla f(\boldsymbol{\gamma}(t)) \cdot \boldsymbol{\gamma}'(t)$. Or, puisque $g(t) = c$, nous en déduisons que : $\nabla f(\boldsymbol{\gamma}(t_0)) \cdot \boldsymbol{\gamma}'(t_0) = 0$, en posant $\boldsymbol{\gamma}(t_0) = \mathbf{x}_0$. La traduction de cette dernière relation est que le gradient de $f$ en $\mathbf{x}_0$ est perpendiculaire au vecteur $\boldsymbol{\gamma}'(t_0)$. Si nous balayons l'ensemble des surfaces de niveau de type $\Gamma$, le gradient reste perpendiculaire à tout vecteur tangent à ces courbes. Si nous supposons que $\nabla f(\boldsymbol{\gamma}(t_0)) \neq \mathbf{0}$, alors l'ensemble de ces vecteurs tangents défini le plan tangent à la surface de niveau $\mathcal{L}(c)$ en $\mathbf{x}_0$. Une caractérisation simple du plan tangent en $\mathbf{x}_0$ est donnée par l'ensemble des points $\mathbf{x} \in \mathbb{R}^3$ satisfaisant l'équation $\nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) = 0$.

## 2.2 Maxima, minima et points-selle (à cheval sur l'optimisation)

Une surface qui est décrite par une équation $z = f(x, y)$ peut être pensée comme une surface de niveau en posant $F(x, y, z) = f(x, y) - z$. Si nous supposons $f$ différentiable, nous avons

$$\nabla F = \partial_x f \mathbf{i} + \partial_y \mathbf{j} - z\mathbf{k},$$

qui conduit, par simple dérivation, à l'équation du plan tangent en un point donné $\mathbf{x}_0 = (x_0, y_0, z_0)$

$$z - z_0 = A(x - x_0) + B(y - y_0),$$

en posant $A = \partial_x f(x_0, y_0)$ et $B = \partial_y f(x_0, y_0)$. Lorsque $\nabla f(\mathbf{x}_0) = \mathbf{0}$ (c'est-à-dire lorsque $A = B = 0$), le point $\mathbf{x}_0$ est appelé point stationnaire ou critique de $f$. Dans la situation que nous considérons, nous avons alors en un point stationnaire $\nabla F(\mathbf{x}_0) = -\mathbf{k}$, ce qui signifie géométriquement que le plan tangent est horizontal en ce point. Afin de distinguer les différents types de points stationnaires, nous utilisons la classification suivante qui est très communément pratiquée : maxima, minima et points-selle. Si nous imaginons que la surface considérée est un paysage montagneux, cette classification revient à séparer les sommets de montagnes, les fonds de vallées et finalement les passages montagneux. Nous appelons, indistinctement extremum un minimum ou un maximum du champ $f$. Il est crucial en pratique de faire le distinguo entre extremum global ou local (relatif).

**Définition 10** *Un point $\mathbf{x}_0$ est appelé minimum absolu de $f$ sur un ensemble $C$ de $\mathbb{R}^n$ si*

$$\forall x \in C, \quad f(\mathbf{x}_0) \leq f(\mathbf{x}).$$

*La fonction $f$ possède un minimum local (ou relatif) sur $C$ en $\mathbf{x}_0$ si l'inégalité précédente est satisfaite dans un voisinage de $\mathbf{x}_0$, c'est-à-dire pour des points $\mathbf{x}$ se trouvant dans des boules $\mathcal{B}(\mathbf{x}_0)$. Finalement, l'extension de l'appellation à un maximum est directe.*

Si nous considérons un extremum de $f$ en un point $\mathbf{x}_0$ où $f$ est différentiable, alors : $\nabla f(\mathbf{x}_0) = \mathbf{0}$. Dans le cas d'une surface, cela signifie que le plan tangent est horizontal en $(\mathbf{x}_0, f(\mathbf{x}_0))$. Toutefois, la condition sur le gradient n'est pas suffisante bien sûr. C'est ce qui arrive lorsque notamment nous considérons un point-selle.

**Définition 11** *Supposons $f$ de classe $\mathcal{C}^1$ et soit $\mathbf{x}_0$ un point stationnaire de $f$. Alors, il est appelé point-selle si pour toute boule $\mathcal{B}(\mathbf{x}_0)$ il existe des points $\mathbf{x}$ tels que $f(\mathbf{x}) < f(\mathbf{x}_0)$ et d'autres tels que $f(\mathbf{x}) > f(\mathbf{x}_0)$.*

Dans le cas unidimensionnel, nous retrouvons le problème de points d'inflexion.

**Exemple 7** *Considérons la surface d'équation $z = f(x, y) = 2 - x^2 - y^2$ (appelée paraboloïde de révolution). Les courbes de niveau sont des cercles et puisque $f(x, y) \leq f(0, 0)$, $\forall (x, y) \in \mathbb{R}^2$, l'orgine est un point de maximum absolu. Un exemple similaire mais pour un point de minimum absolu revient à "renverser la coupe" et à considérer par exemple la surface $z = x^2 + y^2$.*

**Exemple 8** *Considérons la surface d'équation $f(x,y) = xy$ (appelée paraboloïde hyperbolique). Nous avons un point-selle à l'origine. En effet, le point $\mathbf{x}_0 = \mathbf{0}$ est stationnaire puisque $\nabla f(\mathbf{x}_0) = \mathbf{0}$. Soit par exemple $(x,y) \in \mathbb{R}^{*+} \times \mathbb{R}^{*+}$ (ou $(x,y) \in \mathbb{R}^{*-} \times \mathbb{R}^{*-}$), alors $f(x,y) > 0$. Par contre, si nous considérons un point du deuxième ou quatrième quadrant, c'est-à-dire tel que $(x,y) \in \mathbb{R}^{*-} \times \mathbb{R}^{*+}$ ou $(x,y) \in \mathbb{R}^{*+} \times \mathbb{R}^{*-}$, nous avons immédiatement que $f(x,y) < 0$. Ceci correspond bien alors, selon la classification introduite ci-dessus, à dire que $\mathbf{x}_0$ est un point-selle.*

**Exercice 10** *Montrer que l'origine est un point stationnaire pour les trois fonctions suivantes. Préciser alors le type de ce point stationnaire.*

1. $z = f(x,y) = x^3 - 3xy^2$.
2. $z = f(x,y) = x^2y^2$.
3. $z = f(x,y) = 1 - x^2$.

## 2.3 La formule de Taylor au second ordre pour les champs scalaires (un petit effort...)

Supposons que $f$ soit un champ différentiable qui admet un point stationnaire $\mathbf{x}_0$. Il est clair, d'après la classification précédente, que la nature du point stationnaire est définie par le signe de $f(\mathbf{x}_0+\mathbf{y}) - f(\mathbf{x}_0)$, pour un point $\mathbf{x}$ voisin de $\mathbf{x}_0$. La formule de Taylor au premier ordre (cf. l'équation (1.7)) donne

$$f(\mathbf{x}_0 + \mathbf{y}) - f(\mathbf{x}_0) = \|\mathbf{y}\|\, E(\mathbf{x}_0; \mathbf{y}),$$

où $\lim_{\mathbf{y}\to 0} E(\mathbf{x}_0; \mathbf{y}) = 0$. Nous voyons clairement que nous ne possédons pas suffisamment d'informations pour conclure sur la nature de ce point. Il faut alors pousser plus loin le développement de Taylor.

Supposons que $f$ soit deux fois continuement différentiable et soit la forme quadratique $\mathcal{H}_{\mathbf{x}}$ définie en un point $\mathbf{x}$ par

$$\forall \mathbf{y} \in \mathbb{R}^n, \quad \mathcal{H}_{\mathbf{x}}(\mathbf{y},\mathbf{y}) = H(\mathbf{x})\mathbf{y}\cdot\mathbf{y}, \tag{2.1}$$

où $H(\mathbf{x})$ est la matrice des dérivées secondes de $f$ au point $\mathbf{x}$ : $H(\mathbf{x}) = \left[\partial_{x_i,x_j} f(\mathbf{x})\right]_{i,j=1}^{n,n}$, appelée matrice hessienne (on la note quelques fois $D^2 f$). Sous les hypothèses de régularité ci-dessus, c'est une matrice symétrique. On peut récrire (2.1) sous la forme

$$\forall \mathbf{y} = (y_1, ..., u_n) \in \mathbb{R}^n, \quad H(\mathbf{x})\mathbf{y}\cdot\mathbf{y} = \sum_{i,j=1}^{n} \partial_{x_i,x_j} f(\mathbf{x}) y_i y_j.$$

On a alors la formule de Taylor au second ordre pour les champs scalaires.

**Théorème 13** *Soit $f$ un champ scalaire qui admet des dérivées partielles secondes $\left(\partial_{x_i,x_j} f(\mathbf{x})\right)_{i,j=1}^{n,n}$ continues dans une boule $\mathcal{B}(\mathbf{x}_0)$. Alors, pour tout $\mathbf{y} \in \mathbb{R}^n$ tel que $(\mathbf{x}_0 + \mathbf{y}) \in \mathcal{B}(\mathbf{x}_0)$, nous avons la formule de Taylor au second ordre*

$$f(\mathbf{x}_0 + \mathbf{y}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)\cdot\mathbf{y} + \frac{1}{2} H(\mathbf{x}_0 + \theta\mathbf{y})\mathbf{y}\cdot\mathbf{y}, \tag{2.2}$$

*où* $0 < \theta < 1$, *que l'on peut également récrire sous la forme*

$$f(\mathbf{x}_0 + \mathbf{y}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot \mathbf{y} + \frac{1}{2} H(\mathbf{x}_0)\mathbf{y} \cdot \mathbf{y} + \|\mathbf{y}\|^2 E_2(\mathbf{x}_0; \mathbf{y}), \tag{2.3}$$

*où* $\lim_{\mathbf{y} \to 0} E_2(\mathbf{x}_0; \mathbf{y}) = 0$.

*Preuve.* On vérifie bien que la relation (2.2) implique (2.3), car on a supposé que les dérivées partielles secondes de $f$ sont continues sur $\mathcal{B}(\mathbf{x}_0)$.
Soit $y \in \mathbb{R}^n$ fixé et $g : \ [0,1] \to \mathbb{R}$ définie par $g(h) = f(x_0 + h.y)$. On vérifie que $g \in \mathcal{C}^1[0,1] \cap \mathcal{C}^2]0,1[$. En utilisant la formule des accroissements finis à l'ordre 2 pour $g$ entre 0 et 1, nous obtenons bien l'existence de $\theta = \theta(y)$ tel que (2.2) ait lieu.

Remarquons que (2.2) peut s'obtenir en supposant simplement que $f \in \mathcal{C}^1(\mathcal{B}(\mathbf{x}_0))$ et $f$ deux fois différentiable sur la boule ouverte. Mais alors (2.2) n'implique pas clairement (2.3). ■

Dans le cas d'un point stationnaire, nous avons

$$f(\mathbf{x}_0 + \mathbf{y}) = f(\mathbf{x}_0) + \frac{1}{2} H(\mathbf{x}_0)\mathbf{y} \cdot \mathbf{y} + \|\mathbf{y}\|^2 E_2(\mathbf{x}_0; \mathbf{y}), \tag{2.4}$$

qui laisse espérer que, pour un point $\mathbf{x}$ suffisamment proche de $\mathbf{x}_0$, nous soyons capables de fixer la nature du point $\mathbf{x}_0$.

En réalité, on peut relier le signe de la forme quadratique au signe du spectre de la matrice hessienne. A cette fin, nous avons besoin de la proposition suivante.

**Proposition 1** *Soit* $A = [a_{i,j}]_{i,j=1,1}^{n,n}$ *une matrice réelle symétrique et soit la forme quadratique* $\mathcal{A}$ *associée définie par la relation* $\mathcal{A}(\mathbf{y}, \mathbf{y}) = A\mathbf{y} \cdot \mathbf{y}$. *Alors, nous avons*

*i)* $\mathcal{A}(\mathbf{y}, \mathbf{y}) > 0$, $\forall \mathbf{y} \in \mathbb{R}^n$, $\mathbf{y} \neq \mathbf{0} \Leftrightarrow$ *toutes les valeurs propres de A sont positives strictement (et alors A est dite définie positive),*

*ii)* $\mathcal{A}(\mathbf{y}, \mathbf{y}) < 0$, $\forall \mathbf{y} \in \mathbb{R}^n$, $\mathbf{y} \neq \mathbf{0} \Leftrightarrow$ *toutes les valeurs propres de A sont négatives strictement (et alors A est dite définie négative).*

*Preuve.* Puisque $A$ est symétrique elle admet $n$ valeurs propres $\lambda_1, .., \lambda_n$. On note $E_\lambda$ l'espace propre associé à la valeur propre $\lambda$. Si $\lambda_i$ et $\lambda_j$ sont deux valeurs propres distinctes, on vérifie que $E_{\lambda_i}$ et $E_{\lambda_j}$ sont orthogonaux. En utilisant le procédé de Gram-Schmidt sur les espaces propres, on peut alors construire un base orthonormée $\mathcal{V} = \{v_1, .., v_n\}$ de $\mathbb{R}^n$ telle que $A(v_i) = \lambda_i v_i$. Soit un vecteur $y \in \mathbb{R}^n$. On le décompose dans la base $\mathcal{V}$ et les relations i) et ii) s'obtiennent directement. ■

On a alors le théorème suivant qui fait le lien entre la nature des points stationnaires et le spectre de la matrice hessienne.

**Théorème 14** *Soit* $f$ *un champ scalaire tel que toutes ses dérivées partielles secondes* $\partial_{x_i,x_j} f(\mathbf{x})$, *pour* $1 \leq i, j \leq n$, *soient continues dans une boule* $\mathcal{B}(\mathbf{x}_0)$, *et soit* $H(\mathbf{x}_0)$ *la matrice hessienne en un point stationnaire*

$\mathbf{x}_0$. *Alors,*

*i) si toutes les valeurs propres de* $H(\mathbf{x}_0)$ *sont positives strictement,* $f$ *a un minimum relatif en* $\mathbf{x}_0$,

*ii) si toutes les valeurs propres de* $H(\mathbf{x}_0)$ *sont négatives strictement,* $f$ *a un maximum relatif en* $\mathbf{x}_0$,

*iii) si* $H(\mathbf{x}_0)$ *possède au moins une valeur propre strictement positive et une valeur propre strictement négative, alors* $f$ *a un point-selle en* $\mathbf{x}_0$.

*Preuve.* Avec les hypothèses faites, la matrice hessienne $H(\mathbf{x}_0)$ est symétrique. Soit $\mathcal{V}$ la base orthonormée construite dans la preuve du théorème précédent. On considère le développement limité à l'ordre 2 de $f$ en $x_0$ (c.f. 2.3), et on décompose $y$ dans $\mathcal{V}$. Les propriétés i) à iii) s'obtiennent alors facilement ■

**Remarque 2** *Le thèorème ci-dessus ne couvre pas toutes les possibilités. Par exemple, si on suppose seulement que toutes les valeurs propres sont positives, on ne peut pas conclure directement. Dans ce cas il faut approfondir l'étude.*
*Pour s'en convaincre on pourra reprendre les exemples 7 et 8 ainsi que l'exercice 10*

# Chapitre 3

# Généralités et étude théorique des problèmes d'optimisation

## 3.1 Introduction

On s'intéresse dans ce cours aux problèmes du type suivant : "trouver le minimum d'une fonction sans ou avec contrainte(s)". D'un point de vue mathématique, le problème se formule de la façon suivante :

- problème sans contrainte :

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

- problème avec contrainte :

$$\min_{\mathbf{x}\in C} J(\mathbf{x}).$$

Le problème peut être également posé en dimension infinie. il s'agit dans ce cas de minimiser une fonctionnelle $J$ sur un espace vectoriel de dimension *infinie* (un espace hilbertien par exemple mais non exclusivement). Toutefois, nous adopterons, dans ce cours élémentaire, le point de vue qui consiste à traiter des problèmes d'optimisation en dimension *finie*. La raison principale est, qu'en pratique, on discrétise le problème continu, ce qui consiste à faire une projection sur un espace de dimension finie. En plus d'introduire les notations et notions de l'optimisation, nous donnons également quelques résultats théoriques sur l'optimisation sans ou avec contraintes.

Remarquons tout d'abord que

- maximiser $J$ est équivalent à minimiser $-J$,
- il est important de distinguer entre minimum local et global.

Les contraintes sont souvent de type inégalités

$$C = \{\mathbf{x} \in \mathbb{R}^n \text{ tels que : } \varphi_i(\mathbf{x}) \leq 0, \forall i \in I\},$$

ou égalités

$$C = \{\mathbf{x} \in \mathbb{R}^n \text{ tels que : } \varphi_i(\mathbf{x}) = 0, \forall i \in I\},$$

les fonctions $\varphi_i$ étant des fonctions continues (au moins) de $\mathbb{R}^n$ dans $\mathbb{R}$. Introduisons maintenant la définition suivante.

**Définition 12** *Soit une contrainte inégalité $\varphi_i(\mathbf{x}) \leq 0$ et $\mathbf{x}_0$ un point de $\mathbb{R}^n$. Si $\mathbf{x}_0$ satisfait $\varphi_i(\mathbf{x}_0) < 0$, on dit que la contrainte est inactive en $\mathbf{x}_0$. Si $\mathbf{x}_0$ satisfait $\varphi_i(\mathbf{x}_0) = 0$, on dit que la contrainte est active ou saturée en $\mathbf{x}_0$.*

On rencontre parfois une classification des problèmes d'optimisation, ce qui permet de choisir un algorithme de résolution adapté. Par exemple,on parle

- de programmation linéaire lorsque $J$ est linéaire et $C$ est un polyèdre (en général convexe) défini par

  $$C = \{\mathbf{x} \in \mathbb{R}^n, B\mathbf{x} \leq \mathbf{b}\},$$

  où $B$ est une matrice $m \times n$ et $\mathbf{b}$ un vecteur de $\mathbb{R}^n$,
- de programmation quadratique lorsque $J$ est une fonctionnelle quadratique

  $$J(\mathbf{x}) = \frac{1}{2} A\mathbf{x} \cdot \mathbf{x} + \mathbf{b} \cdot \mathbf{x},$$

  où $A$ est une matrice symétrique définie positive, $C$ étant encore en général un polyèdre convexe,
- de programmation convexe lorsque la fonctionnelle $J$ et l'ensemble $C$ sont convexes ($C$ étant encore polyédrique),
- d'optimisation différentiable lorsque $J$ et les fonctions $\varphi_i$ sont une ou deux fois différentiables.

## 3.2 Résultats d'existence

La plupart des théorèmes d'existence de minimum sont des variantes du théorème classique suivant : une fonction continue sur un compact admet un minimum. Commençons par le théorème suivant.

**Théorème 15** *Soit $J$ une fonction continue sur un sous-ensemble $C$ fermé de $\mathbb{R}^n$. On suppose que*

- *ou bien $C$ est borné,*
- *ou bien $C$ et non borné et $\lim_{\|\mathbf{x}\| \to +\infty} J(\mathbf{x}) = +\infty$ (on dit alors que $J$ est coercive),*

*alors $J$ possède un minimum sur $C$.*

*Preuve.*
si i) a lieu $C$ est compact et la conclusion est alors évidente.
Si ii) a lieu on peut par exemple prouver le résultat comme suit (voir aussi exercice 3.6). Soit $(x_n)$ une suite minimisante pour $J$, c'est à dire une suite telle que

$$J(x_n) \underset{n \to \infty}{\to} \inf_{x \in C} J(x). \tag{3.1}$$

Puisque $J$ est coercive, on vérifie que $(x_n)$ est bornée et on peut donc en extraire une sous suite $(x_{n_k})$ qui converge vers un certain élément $x^*$. Puisque $J$ est continue, $J(x_{n_k})$ converge vers $J(x^*)$. On déduit alors de (3.1) que $J(x^*) = \inf_{x\in C} J(x)$. ■

## 3.3 Convexité

La convexité joue un rôle extrêmement important en optimisation. Donnons quelques définitions.

**Définition 13** *Un ensemble $C$ est dit convexe si, pour tous points $\mathbf{x}$ et $\mathbf{y}$ de $C$, le segment $[\mathbf{x};\mathbf{y}]$ est inclus dans $C$, i.e., $\forall t \in [0;1]$, $t\mathbf{x} + (1-t)\mathbf{y}$ est un point de $C$. Une fonction $J$ définie sur un ensemble convexe $C$ est dite convexe si*

$$\forall(\mathbf{x},\mathbf{y}) \in C \times C, \quad \forall t \in [0;1], \quad J(t\mathbf{x} + (1-t)\mathbf{y}) \leq tJ(\mathbf{x}) + (1-t)J(\mathbf{y}).$$

*La fonction est dite strictement convexe si*

$$\forall(\mathbf{x},\mathbf{y}) \in C \times C, \quad \mathbf{x} \neq \mathbf{y}, \quad \forall t \in ]0;1[, \quad J(t\mathbf{x} + (1-t)\mathbf{y}) < tJ(\mathbf{x}) + (1-t)J(\mathbf{y}).$$

Lorsqu'une fonction convexe est dérivable, la caractérisation suivante sera utile.

**Proposition 2** *Soit $J$ une fonction différentiable définie sur un convexe $C$ de $\mathbb{R}^n$, alors $J$ est convexe si et seulement si*

$$\forall(\mathbf{x},\mathbf{y}) \in C \times C, \quad \nabla J(\mathbf{x}) \cdot (\mathbf{y} - \mathbf{x}) \leq J(\mathbf{y}) - J(\mathbf{x}).$$

*Preuve.*
• Soit $J$ convexe. On considère $\theta \in [0,1]$, et $x, y \in C$. Puisque $C$ est convexe $x + \theta(y-x) \in C$ et en utilisant la convexité de $J$ il vient : $J(x + \theta(y-x)) \leq (1-\theta)J(x) + \theta J(y)$. D'où :

$$J(y) - J(x) \geq \frac{J(x + \theta(y-x)) - J(x)}{\theta}. \tag{3.2}$$

On conclut alors en faisant tendre $\theta$ vers 0 dans (3.2).
• Réciproquement, on part des deux inégalités :

$$\begin{aligned} J(y) &\geq J(y + \theta(x-y)) - \theta J(y + \theta(x-y))(x-y) \\ J(x) &\geq J(y + \theta(x-y)) + (1-\theta)J(y + \theta(x-y))(x-y). \end{aligned}$$

En combinant ces deux inégalités on obtient : $J(x+(1-\theta)y) \leq \theta J(x) + (1-\theta)J(y)$, ce qui établit la convexité. ■

**Remarque 3** *On a une caractérisation similaire pour les fonctions strictement convexe : remplacer convexe par strictement convexe et l'inégalité large par une inégalité stricte dans la proposition 2.*

*Toutefois la premier sens de la preuve ne peut pas être directement adaptée. En effet, si l'on obtient bien (3.2) avec une inégalité stricte, cette dernière n'est pas nécéssairement conservée lorsque $\theta$ tend vers 0.*

Le résultat suivant montre l'impact de la convexité dans les problèmes d'optimisation.

**Proposition 3** *Soit $J$ une fonction convexe définie sur un ensemble convexe $C$. Alors,*
- *tout minimum local de $J$ sur $C$ est un minimum global,*
- *si $J$ est strictement convexe, il y a au plus un minimum global.*

*Preuve.*
i) Soit $x_0 \in C$ un minimum local, i.e. il existe $r > 0$ tel que $B := B(x_0, r) \subset C$ et

$$\forall x \in B \quad J(x) \geq J(x_0). \tag{3.3}$$

Soit $y \in C$. On considère $g$ définie par

$$g(t) = J(ty + (1-t)x_0). \tag{3.4}$$

Soit $z = \partial B \cap [xy]$. Il existe $\epsilon \in [0,1]$ tel que $z = \epsilon y + (1-\epsilon)x_0$. La relation (3.3) implique que

$$\forall t \in [0, \epsilon] \quad g(t) \geq J(x_0).$$

En utilisant cette dernière inégalité et le caractère convexe de $J$ nous obtenons :

$$\forall t \in [0, \epsilon] \quad J(x_0) \leq g(t) \leq tJ(y) + (1-t)J(x_0),$$

d'où : $t(J(y) - J(x_0)) \geq 0 \quad \forall t \in [0, \epsilon]$, et en prenant $t = \epsilon$ on obtient $J(y) \geq J(x_0)$.

ii) Soit $J$ strictement convexe. On suppose que $J$ admet deux minimums globaux distincts $x, y$. Soit alors $t \in ]0,1[$ et $z = tx + (1-t)y$. On aboutit a la contradiction suivante : $J(z) < tJ(x) + (1-t)J(y) = \min_C J$ ■

## 3.4 Conditions d'optimalité

Nous supposons dans tout ce paragraphe que $J$ est un ou deux fois différentiable. On notera $\mathbf{x}^*$ un minimum (local) de $J$.

### 3.4.1 Cas sans contraintes

Ce qui suit reste valable dans le cas où le minimum $\mathbf{x}^*$ se trouve à l'intérieur de l'ensemble des contraintes. Nous donnons les conditions nécessaires de minimum, puis celles suffisantes.

**Théorème 16** *Conditions nécessaires.*
*Les deux conditions nécessaires sont les suivantes*

- *Condition au premier ordre : si $J$ est différentiable en $\mathbf{x}^*$, on a $\nabla J(\mathbf{x}^*) = \mathbf{0}$,*
- *Condition au second ordre : si $J$ est deux fois différentiable au point $\mathbf{x}^*$, alors la forme quadratique $D^2J(\mathbf{x}^*)$ est positive i.e.*

$$\left\langle D^2J(\mathbf{x}^*)\mathbf{y}, \mathbf{y}\right\rangle \geq 0,$$

*où $D^2J(\mathbf{x}^*)$ est la matrice hessienne, définie par les coefficients $\dfrac{\partial^2 J}{\partial_{x_i}\partial_{x_j}}(\mathbf{x}^*)$).*

*Preuve.*
i) on considère un développement limité à l'ordre un en $x^*$ :

$$J(x^* + h) = J(x^*) + \nabla J(x^*)h + \|h\|E(x^*; h).$$

Soit $t \geq 0$. On choisit $h = -t\nabla J(x^*)$ et on pose $\alpha = \|\nabla J(x^*)\|$. Il vient

$$J(x^* + h) = J(x^*) - t\alpha^2 + t\alpha E(x^*; h).$$

Supposons que $\alpha > 0$. Puisque $E(x^*; h) \underset{t\to 0}{\to} 0$ on vérifie alors qu'il existe $t^* > 0$ tel que
$\forall t \leq t^* : \quad J(x^* + h) < J(x^*)$. Ceci indique que $x^*$ n'est pas un minimum. Ainsi il est nécéssaire que $\alpha = 0$.

ii) Supposons qu'il existe $y \in \mathbb{R}^n, y \neq 0$ tel que $a := \langle D^2J(\mathbf{x}^*)\mathbf{y}, \mathbf{y}\rangle < 0$. On considère un développement limité à l'ordre deux en $x^*$ :

$$J(x^* + ty) = J(x^*)\frac{1}{2}t^2a + t^2a^2\alpha E_2(x^*; ty).$$

Puisque $E_2(x^*; ty) \underset{t\to 0}{\to} 0$, on vérifie qu'il existe $t^* > 0$ tel que $\forall t \in [0, t^*] : \quad J(x^* + ty) < J(x^*)$. Ceci indique que $x^*$ n'est pas un minimum. Ainsi il est nécéssaire que $D^2J(\mathbf{x}^*)$ soit positive. ■

Nous énonçons à présent des conditions nécéssaires et suffisantes pour qu'un point $x^*$ soit un minimum, dans le cas où $J$ est suffisamment régulière.

**Théorème 17** *Conditions nécéssaires et suffisantes.*
*Soit $J$ une fonction de classe $\mathcal{C}^1$ définie sur $\mathbb{R}^n$. On suppose que : $\nabla J(\mathbf{x}^*) = \mathbf{0}$ et que $J$ est deux fois différentiable en $x^*$.*
*Alors, $\mathbf{x}^*$ est un minimum (local) de $J$ si et seulement si l'une des deux conditions suivantes est vérifiée*

*i) $D^2J(\mathbf{x}^*)$ est définie positive,*
*ii) $\exists r > 0$ tel que $J$ est deux fois différentiable sur $B(x^*, r)$ et, la forme quadratique $D^2J(\mathbf{x})$ est positive pour tout $x \in B(x^*, r)$*

*Preuve.*
• La condition nécéssaire au premier et au deuxième ordre en $x^*$ est vérifiée dans le deux cas i) et ii).
• On vérifie que i) ou ii) est une condition suffisante. Pour i) : voir théorème 14. Pour ii) : on considère la

formule de Taylor-MacLaurin en $x^*$ à l'ordre 2. Pour tout $h \in B(0,r)$ il existe $\lambda = \lambda(h) \in (0,1)$ tel que

$$J(x^* + h) = J(x^*) + \frac{1}{2}\langle D^2 J(\mathbf{x}^* + \lambda h)\mathbf{h}, \mathbf{h}\rangle \geq J(x^*),$$

ce qui montre bien que $x^*$ est un minimum. ■

Dans le cas où $J$ est convexe, la condition suffisante s'exprime beaucoup plus facilement. En effet, nous avons la

**Proposition 4** *Soit $J$ une fonction convexe de classe $\mathcal{C}^1$, définie sur $\mathbb{R}^n$ et $\mathbf{x}^*$ un point de $\mathbb{R}^n$. Alors, $\mathbf{x}^*$ est un minimum (global) de $J$ si et seulement si $\nabla J(\mathbf{x}^*) = \mathbf{0}$.*

*Preuve.*
Il suffit de montrer que la condition est suffisante. Soit $y$ quelconque. D'après la proposition 2 nous avons :

$$J(y) - J(x^*) \geq \nabla J(x^*)(y - x^*) = 0,$$

ce qui montre bien que $x^*$ est un minimum. ■

### 3.4.2 Cas avec contraintes

Dans cette second situation, plus complexe, nous avons le résultat suivant.

**Proposition 5** *Soit $J$ une fonction convexe de classe $\mathcal{C}^1$, définie sur un ensemble convexe $C \subseteq \mathbb{R}^n$, et $\mathbf{x}^*$ un point de $C$. Alors, $\mathbf{x}^*$ est un minimum (global) de $J$ sur $C$ si et seulement si*

$$\forall \mathbf{y} \in C, \quad \nabla J(\mathbf{x}^*) \cdot (\mathbf{y} - \mathbf{x}) \geq 0.$$

#### 3.4.2.1 Contraintes inégalités

On suppose, dans cette partie, que l'ensemble $C$ sur lequel on veut minimiser $J$ est donné par des contraintes de type inégalités

$$C = \{\mathbf{x} \in C, g_i(\mathbf{x}) \leq 0, \forall i \in I = \{1, ..., m\}\},$$

où les $g_i$ sont des fonctions de classe $\mathcal{C}^1$ de $\mathbb{R}^n$ dans $\mathbb{R}$.

**Définition 14** *On dit qu'un arc de courbe $\boldsymbol{\gamma} : [0;\varepsilon] \to \mathbb{R}^n$ est admissible si $\boldsymbol{\gamma}(0) = \mathbf{x}^*$ et $\boldsymbol{\gamma}(t) \in C$, $\forall t > 0$ suffisamment voisin de 0. On appelle direction admissible au point $\mathbf{x}^*$ les vecteurs tangents au point $\mathbf{x}^*$ des courbes admissibles. On note $C_{ad}(\mathbf{x}^*)$ le cône des directions admissibles au point $\mathbf{x}^*$.*

Nous notons $I_0$ (sous-entendu $I_0(\mathbf{x}^*)$ l'ensemble des contraintes saturées au point $\mathbf{x}^*$, c'est-à-dire l'ensemble

$$I_0 = \{i \in I, \quad g_i(\mathbf{x}^*) = 0\}.$$

On a alors les deux propriétés suivantes.

**Proposition 6** *Si* $\mathbf{y}$ *est une direction admissible au point* $\mathbf{x}^*$*, alors nous avons l'inégalité suivante sur les contraintes saturées*

$$\forall i \in I_0, \quad \nabla g_i(\mathbf{x}^*) \cdot \mathbf{y} \leq 0. \tag{3.5}$$

De plus, nous avons la seconde proposition.

**Proposition 7** *Si* $\mathbf{x}^*$ *est un minimum de la fonction* $J$*, alors*

$$\nabla J(\mathbf{x}^*) \cdot \mathbf{y} \geq 0,$$

*pour toute direction admissible* $\mathbf{y}$*. Cette dernière inégalité porte aussi le nom d'inégalité d'Euler.*

Afin de préciser les conditions d'optimalité, il est nécessaire de décrire plus précisément les directions admissibles. La relation (3.5) ne le permet pas puisque ce n'est pas une équivalence. Introduisons la définition suivante.

**Définition 15** *Nous dirons que les contraintes sont qualifiées au point* $\mathbf{x}^*$ *si*

- *ou bien les fonctions* $g_i$ *sont affines,*
- *ou bien les vecteurs* $\nabla g_i(\mathbf{x}^*)$*,* $i \in I_0$*, sont linéairement indépendants.*

Nous avons alors la caractérisation.

**Proposition 8** *Si les contraintes sont qualifiées au point* $\mathbf{x}^*$*, alors* $\mathbf{y}$ *est une direction admissible si et seulement si*

$$\nabla g_i(\mathbf{x}^*) \cdot \mathbf{y} \leq 0, \quad \forall i \in I_0.$$

On a alors le théorème fondamental suivant dû à Kuhn-Tucker.

**Théorème 18** *(Kuhn-Tucker (1951)). Soit* $J$ *et* $g_i$*,* $i \in I = \{1, ..., m\}$*, des fonctions de classe* $\mathcal{C}^1$*. On suppose les contraintes qualifiées au point* $\mathbf{x}^*$*. Alors, une condition nécessaire pour que* $\mathbf{x}^*$ *soit un minimum de* $J$ *sur l'ensemble* $C = \{\mathbf{x} \in \mathbb{R}^n, g_i(\mathbf{x}) \leq 0, i \in I\}$*, est qu'il existe des nombres positifs* $\lambda_1, ..., \lambda_m$ *(appelés multiplicateurs de Kuhn-Tucker ou de Lagrange généralisés) tels que*

$$\begin{cases} \nabla J(\mathbf{x}^*) + \displaystyle\sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}, \\ \text{avec} \quad \lambda_i g_i(\mathbf{x}^*) = 0, \quad \forall i \in I. \end{cases}$$

Attention, ce résultat n'est pas une équivalence au sens ou ce n'est pas une condition nécessaire et suffisante de caractérisation de minimum. Elle le devient par contre lorsque nous considérons une fonction convexe. En effet, nous avons le

**Théorème 19** *On reprend les hypothèses du thórème de Kuhn-Tucker et on suppose de plus que $J$ et les $g_i$ sont convexes. Alors, $\mathbf{x}^*$ est un minimum de $J$ sur $C = \{\mathbf{x} \in \mathbb{R}^n, g_i(\mathbf{x}) \leq 0, i \in I\}$ si et seulement si il existe des nombres positifs $\lambda_1, ..., \lambda_m$ tels que*

$$\begin{cases} \nabla J(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}, \\ avec \quad \lambda_i g_i(\mathbf{x}^*) = 0, \quad \forall i \in I. \end{cases}$$

#### 3.4.2.2 Contraintes égalités

Nous supposons ici que l'ensemble des contraintes est de type égalités, i.e. $f_i(\mathbf{x}) = 0$. Puisqu'une contrainte égalité est équivalente à deux contraintes inégalités, à savoir $f_i(\mathbf{x}) \leq 0$ et $-f_i(\mathbf{x}) \leq 0$, nous allons pouvoir nous ramener au cas précédent. Ce qu'il faut retenir dans ce cas est que

- les contraintes sont forcément saturées (évident),
- pour qu'une direction $\mathbf{y}$ soit admissible, il faut supposer ici $\nabla f_i(\mathbf{x}^*) \cdot \mathbf{y} = 0$, pour tout $i$,
- on a la même notion de contraintes qualifiées : si on suppose que les vecteurs $\nabla f_i(\mathbf{x}^*)$ sont linéairement indépendants, alors $\mathbf{y}$ est admissible si et seulement si : $\nabla f_i(\mathbf{x}^*) \cdot \mathbf{y} = 0$, pour tout $i$.

Lorsque l'on écrit la condition de Kuhn-Tucker pour les contraintes égalités, nous allons avoir

$$\nabla J(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^1 \nabla f_i(\mathbf{x}^*) - \lambda_i^2 \nabla f_i(\mathbf{x}^*) = \mathbf{0},$$

ou encore

$$\nabla J(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i \nabla f_i(\mathbf{x}^*) = \mathbf{0},$$

en posant $\mu_i = \lambda_i^1 - \lambda_i^2$. Les multiplicateurs $\mu_i$ ne vérifient pas de conditions de signes (contrairement au cas avec contraintes inégalités). Ces nombres s'appellent multiplicateurs de Lagrange.

Afin de résumer l'ensemble de ces résultats, énonçons le théorème suivant.

**Théorème 20** *(Kuhn-Tucker, Lagrange). Soit $J$, $f_i$, $i \in \{1, ..., p\}$, $g_i$, $i \in \{1, ..., m\}$, des fonctions de classe $\mathcal{C}^1$. On veut minimiser $J$ sur l'ensemble*

$$C = \{\mathbf{x} \in \mathbb{R}^n, f_i(\mathbf{x}^*) = 0, \quad i \in \{1, ..., p\}, \quad g_i(\mathbf{x}^*) \leq 0, \quad i \in \{1, ..., m\}\}.$$

*On suppose les contraintes qualifiées au point $\mathbf{x}^*$. Alors, une condition nécessaire pour que $\mathbf{x}^*$ soit un minimum de $J$ est qu'il existe des nombres positifs $\lambda_1, ..., \lambda_m$, et des nombres réels $\mu_1, ..., \mu_p$, tels que*

$$\begin{cases} \nabla J(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*) = \sum_{i=1}^{p} \mu_i \nabla f_i(\mathbf{x}^*), \\ avec \quad \lambda_i g_i(\mathbf{x}^*) = 0, \quad 1 \leq i \leq m. \end{cases}$$

## 3.5 Deux exemples qui permettent de mieux saisir ce que sont les multiplicateurs de Lagrange

### 3.5.1 Le premier problème

On se donne une surface $S$ ne passant pas par l'origine. On se propose de déterminer les points $\mathbf{x}^*$ de $S$ les plus proches de l'origine.

Comment poser le problème ? La fonction à minimiser ici est la fonction distance

$$J(\mathbf{x}) = \|\mathbf{x}\|\,,$$

et l'ensemble des contraintes $C$ est la surface elle-même puisque l'on doit en effet avoir $\mathbf{x}^* \in S$. On s'attaque donc bien à un problème sous contraintes du type

$$\min_{\mathbf{x}\in C} J(\mathbf{x}).$$

Désignons par $r$ la distance d'un point $\mathbf{x}$ à l'origine. Alors, un point est à distance $r$ de l'origine s'il satisfait : $\|\mathbf{x}\| = r$, c'est-à-dire s'il se trouve sur la sphère de centre l'origine et de rayon $r$. Commençons à $r = 0$ puis augmentons la valeur de $r$. A un moment donné, cette surface de niveau va toucher $S$, chaque point de contact $\mathbf{x}^*$ étant alors un point que nous cherchons. Pour déterminer les coordonnées du point de contact, nous supposons que $S$ est décrite par une équation cartésienne : $f_1(\mathbf{x}) = 0$. Si maintenant $S$ a un plan tangent en un point de contact, ce plan tangent doit être également tangent à la surface de niveau. En cela, le gradient de la surface $f_1(\mathbf{x}) = 0$ doit être parallèle au gradient de la surface de contact $J(\mathbf{x}) = r$ ; ceci veut dire qu'il existe un scalaire $\mu$ tel que nous ayons : $\nabla J = \mu \nabla f_1$, en tout point de contact. On voit donc que cette dernière équation correspond bien à l'approche par multiplicateurs de Lagrange, $\mu$ étant le multiplicateur.

### 3.5.2 Le second problème

Soit $T(\mathbf{x})$ la température en un point $\mathbf{x}$ de l'espace tridimensionnel. On se pose alors la question de la détermination des maxima et minima de la température sur une courbe $C$ donnée de l'espace $\mathbb{R}^3$.

D'un point de vue de l'optimisation, le problème se pose de la manière suivante

$$\min_{\mathbf{x}\in C} J(\mathbf{x}),$$

en posant $J(\mathbf{x}) = T(\mathbf{x})$. Si nous voyons la courbe $C$ comme l'intersection de deux surfaces, disons $f_1(\mathbf{x}) = 0$ et $f_2(\mathbf{x}) = 0$, nous avons alors un problème d'extrema avec deux contraintes égalités. Les deux vecteurs gradients $\nabla f_1$ et $\nabla f_2$ sont normaux à ces surfaces, et, par conséquent, sont également normaux à la courbe $C$, courbe intersection de ces deux surfaces.

Nous allons montrer que $\nabla J$, le gradient de température, est également normal à $C$. Dans l'immédiat, supposons le. Alors, $\nabla J$ se trouve dans le plan défini par $\nabla f_1$ et $\nabla f_2$ ; ainsi, si $\nabla f_1$ et $\nabla f_2$ sont indépendants,

nous pouvons exprimer $\nabla J$ comme une combinaison linéaire de $\nabla f_1$ et $\nabla f_2$

$$\nabla J = \mu_1 \nabla f_1 + \mu_2 \nabla f_2,$$

qui serait donnée par les multiplicateurs de Lagrange. Pour montrer que $\nabla J$ est normal à $C$ en un extremum, imaginons que $C$ est décrite par une fonction à valeurs vectorielles $\boldsymbol{\alpha}(t)$, où $t$ varie dans un intervalle $I = [a; b]$. Sur la courbe $C$, la température devient une fonction de $t$, disons $g(t) = f(\boldsymbol{\alpha}(t))$. Si $g$ a un extremum relatif en un point intérieur $t^*$, nous devons avoir : $g(t^*) = 0$. Par ailleurs, la règle de dérivation en chaîne nous dit que $g'(t)$ est donné par

$$g'(t) = \nabla J(\boldsymbol{\alpha}(t)) \cdot \boldsymbol{\alpha}'(t).$$

Ce produit est nul si $\nabla J$ est perpendiculaire à $\boldsymbol{\alpha}'(t^*)$. Mais, le vecteur $\boldsymbol{\alpha}'(t^*)$ se trouve dans le plan normal à $C$, ce que nous voulions. Par ailleurs, les deux gradients des contraintes $\nabla f_1$ et $\nabla f_2$ sont linéairement indépendants si et seulement si $\nabla f_1 \wedge \nabla f_2 \neq 0$, ce qui donne une condition d'indépendance.

Il est impératif de remarquer que la méthode des multiplicateurs échoue si $\nabla f_1$ et $\nabla f_2$ sont linéairement dépendants. Par exemple, supposons que nous essayons d'appliquer la méthode à la recherche des valeurs extrêmales de $J(\mathbf{x}) = x^2 + y^2$, sur la courbe définie par l'intersection de deux surfaces $f_1(\mathbf{x}) = 0$ et $f_2(\mathbf{x}) = 0$, où $f_1(\mathbf{x}) = z$ et $f_1(\mathbf{x}) = z^2 - (y-1)^3$. Les deux surfaces, un plan et un cylindre, s'intersectent le long de la ligne $y = 1$. Le problème a bien évidemment comme solution : $\mathbf{x}^* = (0, 1, 0)$. Toutefois, en ce point, $\nabla f_1(\mathbf{x}^*) = \mathbf{k}$ et $\nabla f_2(\mathbf{x}^*) = \mathbf{0}$. Par ailleurs, $\nabla J(\mathbf{x}^*) = 2\mathbf{j}$. Il est donc clair que l'on ne peut écrire $\nabla J(\mathbf{x}^*)$ comme une combinaison linéaire entre $\nabla f_1(\mathbf{x}^*)$ et $\nabla f_2(\mathbf{x}^*)$.

## 3.6 Exercices

**Exercice 3.1**

Les fonctions suivantes sont-elles coercives ?

1. $J(x) = x^3 + x^2 + 1$, définie de $\mathbb{R}$ dans $\mathbb{R}$.
2. $J(\mathbf{x}) = x_1^2 + 2x_2^2 - ax_1 - bx_2 - c$, avec $a$, $b$ et $c$, trois réels.
3. $J(\mathbf{x}) = x_1^2 - x_2^2$.
4. $J(\mathbf{x}) = 2x_1^2 + x_2^3 + 2x_2^2$, définie de $\mathbb{R}^2$ dans $\mathbb{R}$.

**Exercice 3.2**

Soient $x$ et $y$, deux réels strictement positifs. Soit $p$, un nombre entier naturel. On appelle $q$ son conjugué, c'est à dire le nombre vérifiant $\dfrac{1}{p} + \dfrac{1}{q} = 1$.
Démontrer qu'alors, on a : $xy \leq \dfrac{x^p}{p} + \dfrac{y^q}{q}$.
*Indication :* on utilisera pour cela la convexité d'une fonction judicieusement choisie. Cette inégalité porte le nom d'inégalité de Young.

**Exercice 3.3**

Soit $E$, un espace vectoriel de dimension finie, et $f : E \longrightarrow \mathbb{R}$, une application continue et coercive. Soit $F$, un fermé de $E$.

1. Démontrer l'existence de $R > 0$ tel que $\inf_{\mathbf{x}\in F} f(\mathbf{x}) = \inf_{\mathbf{x}\in F\cap\mathcal{B}_f(0,R)} f(\mathbf{x})$, où $\mathcal{B}_f(0, R)$ désigne la boule fermée de centre 0 et rayon $R$.
2. En déduire que $f$ est minorée et qu'elle atteint sa borne inférieure.

**Exercice 3.4**

Soit $A$, une matrice de taille $n \times n$ symétrique et définie positive. Soit $\mathbf{b}$, un vecteur de taille $1 \times n$.

1. Démontrer qu'il existe $\nu > 0$ tel que pour tout vecteur $\mathbf{x}$ de taille $n \times 1$, $(A\mathbf{x}, \mathbf{x}) \geq \nu\|\mathbf{x}\|^2$.
2. Soit $J$, la fonctionnelle définie sur $\mathbb{R}^n$ pour $\mathbf{x} \in \mathbb{R}^{n\times 1}$ par :

$$J(\mathbf{x}) = \frac{1}{2}(A\mathbf{x}, \mathbf{x}) - (\mathbf{b}, \mathbf{x}).$$

   (a) Démontrer que $J$ est coercive.

   (b) En utilisant le résultat démontré dans l'exercice précédent, en déduire, en écrivant les conditions d'optimalité, que le problème

$$\begin{cases} \min J(\mathbf{x}) \\ \mathbf{x} \in \mathbb{R}^{n\times 1}. \end{cases}$$

   possède une unique solution que l'on déterminera complètement.

**Exercice 3.5**

On considère un nuage de points de $\mathbb{R}^2$ tels que $M_i = (x_i, t_i)$, pour $1 \leq i \leq n$. En pratique, ces points résultent de mesures prises lors d'expériences. On cherche alors à déterminer le comportement global de ce nuage de points. Bien sûr, ceux-ci n'ont aucune raison d'être alignés. On décide toutefois de chercher une droite qui les approche au mieux...

On utilise alors ce que l'on appelle la **méthode des moindres carrés**. Puisque l'on n'a pas $x_i = at_i + b$, pour tout indice $i$, on cherche à minimiser la fonctionnelle $J$ définie par $J(a, b) := ||\mathbf{x} - a\mathbf{t} - b||^2$. Nous avons donc à résoudre un problème de minimisation sans contrainte

$$\begin{cases} \min J(a, b) \\ (a, b) \in \mathbb{R}^2 \end{cases}$$

On introduit la notation suivante : $S_{f(\mathbf{x},\mathbf{t})} = \sum_{i=1}^{n} f(x_i, t_i)$.

1. Montrer que $\nabla J(a, b) = 0$ est équivalent à écrire

$$\begin{cases} S_{t^2}a + S_t b = S_{xt} \\ S_t a + nb = S_x \end{cases}$$

2. Si $S_t^2 - nS_{t^2} \neq 0$, donner la solution de ce système.

3. Cette solution est-elle, sous cette condition, un minimum ?

**Exercice 3.6**

Soit $A$, une matrice de taille $n \times n$ symétrique. Soit $\mathbf{b}$, un vecteur de taille $1 \times n$. On appelle $J$, la fonctionnelle définie sur $\mathbb{R}^n$ pour $\mathbf{x} \in \mathbb{R}^{n \times 1}$ par :

$$J(\mathbf{x}) = \frac{1}{2}(A\mathbf{x}, \mathbf{x}) - (\mathbf{b}, \mathbf{x}).$$

1. Calculer la différentielle et la matrice hessienne de la fonctionnelle $J$.
2. En déduire une condition suffisante pour que $J$ soit strictement convexe.

**Exercice 3.7**

1. Pour chacune des fonctions suivantes, définies sur $\mathbb{R}^n$, avec $n$ entier, déterminer les points critiques, puis donner le maximum d'information sur ces points.
   (a) $f(x, y, z) = x^4 + y^2 + z^2 - 4x - 2y - 2z + 4$.
   (b) $f(x, y, z) = x^4 - 2x^2y + 2y^2 + 2z^2 + 2yz - 2y - 2z + 2$.
   (c) $f(x, y) = x^2 + y^2 - xy$.
   (d) $f(x, y) = x^2 - y^2 - xy$.
2. Soit $g$, la fonction définie sur $\mathcal{C}$, le cercle de centre 0 et rayon $R > 0$, par :

$$g(x, y) = x^2 + xy + y^2.$$

   Démontrer que $g$ atteint ses bornes puis les déterminer.

# Chapitre 4

# Quelques algorithmes pour l'optimisation sans contraintes

## 4.1 Introduction

Une grande classe d'algorithmes que nous allons considérer pour les problèmes d'optimisation ont la forme générale suivante

$$\mathbf{x}^{(0)} \text{ étant donné, calculer } \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}. \tag{4.1}$$

Le vecteur $\mathbf{d}^{(k)}$ s'appelle la direction de descente, $\rho^{(k)}$ le pas de la méthode à la $k$-ième itération. En pratique, on s'arrange presque toujours pour satisfaire l'inégalité

$$J(\mathbf{x}^{(k+1)}) \leq J(\mathbf{x}^{(k)}).$$

De tels algorithmes sont souvent appelés méthodes de descente. Essentiellement, la différence entre ces algorithmes réside dans le choix de la direction de descente $\mathbf{d}^{(k)}$. Cette direction étant choisie, nous sommes plus ou moins ramené à un problème unidimensionnel pour la détermination de $\rho^{(k)}$. Pour ces raisons, commençons par analyser ce qui se passe dans le cas de la dimension un.

## 4.2 Algorithmes unidimensionnels ou recherche du pas

Soit $q(\rho)$ la fonction coût que l'on cherche à minimiser. On pourra prendre par exemple

$$q(\rho) = J(\mathbf{x}^{(k)} + \rho\mathbf{d}^{(k)})$$

afin d'appliquer les idées au cas de la méthode de descente. Supposons que l'on connaisse un intervalle $[a; b]$ contenant le minimum $\rho^*$ de $q$ et tel que $q$ soit décroissante sur $[a; \rho^*]$ et croissante sur $]\rho^*; b]$ ($q$ est alors appelée une fonction unimodale).

poser $\tau = \dfrac{1+\sqrt{5}}{2}$
poser $a_0 = a$
poser $b_0 = b$
pour $i = 0, ..., N^{\max}$
    poser $a' = a_i + \dfrac{1}{\tau^2}(b_i - a_i)$
    poser $b' = a_i + \dfrac{1}{\tau}(b_i - a_i)$
    si $(q(a') < q(b'))$ alors
        poser $a_{i+1} = a_i$
        poser $b_{i+1} = b'$
    sinon si $(q(a') > q(b'))$ alors
        poser $a_{i+1} = a'$
        poser $b_{i+1} = b_i$
    sinon si $(q(a') = q(b'))$ alors
        poser $a_{i+1} = a'$
        poser $b_{i+1} = b'$
    fin si
fin pour i

TAB. 4.1 – Algorithme de la section dorée.

### 4.2.1 Méthode de la section dorée

On construit une suite décroissante d'intervalles $[a_i; b_i]$ qui contiennent tous le minimum $\rho^*$. Pour passer de $[a_i; b_i]$ à $[a_{i+1}; b_{i+1}]$, on procède de la manière suivante. On introduit deux nombres $a'$ et $b'$ de l'intervalle $[a_i; b_i]$ et tels que $a' < b'$. Puis, on calcule les valeurs $q(a')$ et $q(b')$. Trois possibilités se présentent alors à nous. Si $q(a') < q(b')$, alors, le minimum $\rho^*$ se trouve nécessairement à gauche de $b'$. Ceci définit alors le nouvel intervalle en posant $a_{i+1} = a_i$ et $b_{i+1} = b'$. Considérons maintenant que l'inégalité : $q(a') > q(b')$ est satisfaite. Dans ce second cas, il est évident que le minimum se trouve cette fois à droite de $a'$. On pose alors : $a_{i+1} = a'$ et $b_{i+1} = b_i$. Enfin, le dernier cas consiste à avoir $q(a') = q(b')$. Alors, le minimum se trouve dans l'intervalle $[a'; b']$. On se restreint donc à $a_{i+1} = a'$ et $b_{i+1} = b'$.

La question suivante se pose : comment choisir $a'$ et $b'$ en pratique ? En général, on privilégie deux aspects :

i) on souhaite que le facteur de réduction $\tau$, qui représente le ratio du nouvel intervalle par rapport au précédent, soit constant,

ii) on désire réutiliser le point qui n'a pas été choisi dans l'itération précédente afin de diminuer les coûts de calculs.

On peut montrer que la vérification simultanée de ces deux contraintes conduit à un choix unique des paramètres $a'$ et $b'$. Plus précisément, supposons que $q$ est unimodale. Alors, on obtient l'algorithme de la table 4.1 dit de la section dorée, la méthode tirant son nom de la valeur du paramètre $\tau$.

Ici, $N^{\max}$ est le nombre maximal d'itérations que l'on se fixe. A cette fin, on doit valider un critère d'arrêt de la forme : $|b_{i+1} - a_{i+1}| < \varepsilon$, où $\varepsilon$ est l'erreur (ou tolérance) que l'on se permet sur la solution $\rho^*$ du problème.

```
choisir x_0, y_0 et z_0 dans [a;b] tels que q(x_0) ≥ q(y_0) et q(z_0) ≥ q(y_0)
pour i = 0, ..., N^max
      poser q[x_i; y_i] = (q(y_i) − q(x_i)) / (y_i − x_i)
      poser q[x_i; y_i; z_i] = (q[x_i; z_i] − q[x_i; y_i]) / (z_i − x_i)
      poser y_{i+1} = (x_i + y_i)/2 − q[x_i; y_i] / (2q[x_i; y_i; z_i])
      si y_{i+1} ∈ [x_i; y_i] alors
            poser x_{i+1} = x_i
            poser z_{i+1} = y_i
      sinon si y_{i+1} ∈ [y_i; z_i] alors
            poser x_{i+1} = y_i
            poser z_{i+1} = z_i
      fin si
fin pour i
```

TAB. 4.2 – Algorithme de l'interpolation parabolique.

### 4.2.2 Méthode d'interpolation parabolique

L'idée maîtresse de la méthode d'interpolation parabolique consiste à remplacer la fonction coût $q$ par son polynôme d"interpolation $p$ d'ordre deux (d'où l'appellation d'interpolation parabolique) en trois points $x_0$, $y_0$ et $z_0$ de l'intervalle $[a;b]$. Ces points sont choisis tels que : $q(x_0) \geq q(y_0)$ et $q(z_0) \geq q(y_0)$. On peut montrer que si nous posons

$$q[x_0; y_0] = \frac{q(y_0) - q(x_0)}{y_0 - x_0},$$

et

$$q[x_0; y_0; z_0] = \frac{q[x_0; z_0] - q[x_0; y_0]}{z_0 - x_0},$$

alors, le minimum est donné par

$$y_1 = \frac{x_0 + y_0}{2} - \frac{q[x_0; y_0]}{2q[x_0; y_0; z_0]}.$$

Il est clair que $\rho^* \in [x_0; z_0]$ selon les choix précédents. On choisit ensuite les trois nouveaux points de la manière suivante

- si $y_1 \in [x_0; y_0]$, on pose alors $x_1 = x_0$, $y_1 = y_1$ et $z_1 = y_0$ puisque $\rho^* \in [x_0; y_0]$,
- si $y_1 \in [y_0; z_0]$, on pose alors $x_1 = y_0$, $y_1 = y_1$ et $z_1 = z_0$ car $\rho^* \in [y_0; z_0]$.

Puis on recommence. Ceci conduit à l'algorithme donné table 4.2.

On peut montrer que la méthode est d'ordre 1.3. En fait, nous pouvons montrer qu'il existe une constante strictement positive $C$, indépendante de $i$, telle que l'on ait l'inégalité

$$|y_{i+1} - \rho^*| \leq C|y_i - \rho^*|^{1.3}.$$

Dire que la méthode est d'ordre 1.3 signifie que si à une étape donnée l'erreur de $10^{-2}$, elle sera de l'ordre de $(10^{-2})^{1.3} \approx 2.5\ 10^{-3}$ à l'étape suivante.

Une des difficultés concerne l'initialisation de l'algorithme. Pratiquement, on peut procéder de la façon suivante. On choisit un point $\alpha_0$ de l'intervalle $[a; b]$ ainsi qu'un pas de déplacement positif $\delta$. On calcule ensuite $q(\alpha_0)$ et $q(\alpha_0 + \delta)$. On a alors deux situations

- si $q(\alpha_0) \geq q(\alpha_0+\delta)$, alors $q$ décroît et donc $\rho^*$ est à droite de $\alpha_0+\delta$. On continue alors à calculer $q(\alpha_0+2\delta)$, $q(\alpha_0 + 3\delta)$,...,$q(\alpha_0 + k\delta)$ jusqu'à tomber sur un entier $k$ tel que $q$ croît : $q(\alpha_0 + k\delta) > q(\alpha_0 + (k-1)\delta)$, avec $k \geq 2$. On pose alors

$$x_0 = \alpha_0 + (k-2)\delta, y_0 = \alpha_0 + (k-1)\delta, z_0 = \alpha_0 + k\delta.$$

- si $q(\alpha_0) < q(\alpha_0 + \delta)$, alors $\rho^*$ est à gauche de $\alpha_0$. On prend $-\delta$ comme pas jusqu'à tomber sur un entier $k$ tel que : $q(\alpha_0 - k\delta) \geq q(\alpha_0 - (k-1)\delta)$. On pose alors

$$x_0 = \alpha_0 - k\delta, y_0 = \alpha_0 - (k-1)\delta, z_0 = \alpha_0 - (k-2)\delta.$$

Ceci permet d'initialiser l'algorithme d'interpolation parabolique en suivant l'algorithme de recherche d'initialisation présenté table 4.3.

### 4.2.3 D'autres règles

La recherche du pas n'est qu'une étape d'un algorithme plus complexe pour minimiser $J$. La philosophie générale est alors de plutôt essayer d'avoir une approximation satisfaisante du pas optimal.

Si nous considérons $q(\rho) = J(\mathbf{x} + \rho\mathbf{d})$, avec $\rho > 0$, nous avons, par application de la règle de dérivation en chaîne (cf. le Théorème 7)

$$q'(\rho) = \nabla J(\mathbf{x} + \rho\mathbf{d}) \cdot \mathbf{d}.$$

Par conséquent, puisque $\mathbf{d}$ est une direction de descente, $q'(0) = \nabla J(\mathbf{x}) \cdot \mathbf{d} < 0$. Il s'ensuit que si nous prenons $\rho$ un peu à droite de 0, on est sûr de faire décroître $q$. Toutefois, il faut faire attention car deux éléments contradictoires sont à prendre en compte

- si $\rho$ est trop grand, on risque de ne pas faire décroître la fonction $q$ ou son comportement peut être oscillant,
- si $\rho$ est trop petit, l'algorithme n'avancera pas assez vite.

Il existe deux règles classiques pour parvenir à cette fin.

#### 4.2.3.1 Règle de Goldstein (1967)

On choisit deux nombres $(m_1, m_2)$ tels que $0 < m_1 < m_2 < 1$ et on recherche une valeur $\rho$ qui vérifie

$$\begin{cases} q(\rho) \leq q(0) + m_1\rho q'(0) \\ q(\rho) \geq q(0) + m_2\rho q'(0) \end{cases}$$

```
choisir α_0 dans [a; b]
choisir δ > 0
poser q_0 = q(α_0)
poser q_1 = q(α_0 + δ)
si (q_1 ≤ q_0) alors
      poser k = 2
      poser q_2 = q(α_0 + kδ)
      tant que (q_2 ≤ q_1) faire
            poser k = k + 1
            poser q_0 = q_1
            poser q_1 = q_2
            poser q_2 = q(α_0 + kδ)
      fin tant que
      poser x_0 = q_0
      poser y_0 = q_1
      poser z_0 = q_2
sinon si (q_1 > q_0) alors
      poser k = 1
      poser q_2 = q(α_0 − kδ)
      tant que (q_2 ≤ q_0) faire
            poser k = k + 1
            poser q_0 = q_2
            poser q_1 = q_0
            poser q_2 = q(α_0 − kδ)
      fin tant que
      poser x_0 = q_2
      poser y_0 = q_0
      poser z_0 = q_1
fin si
```

TAB. 4.3 – Algorithme d'initialisation de l'interpolation parabolique.

choisir $m_1$ et $m_2$ tels que $0 < m_1 < m_2 < 1$
poser $\rho = 1$, $\rho_- = 0$, $\rho_+ = 0$
debut 10 :
    si $q(\rho) \leq q(0) + m_1 \rho q'(0)$ et $q(\rho) \geq m_2 q'(0)$ alors
        poser $\rho^{(k)} = \rho$
        arrêt
    sinon
        si $q(\rho) > q(0) + m_1 \rho q'(0)$ alors
            poser $\rho_+ = \rho$
        sinon si $q(\rho) \leq q(0) + m_1 \rho q'(0)$ et $q(\rho) < m_2 q'(0)$ alors
            poser $\rho_- = \rho$
        fin si
    fin si
fin 10 :
si $\rho_+ = 0$ alors
    choisir $\rho > \rho_-$ (par exemple $\rho = 2\rho_-$)
sinon si $\rho_+ > 0$ alors
    choisir $\rho \in ]\rho_-; \rho_+[$ (par exemple $\rho = \dfrac{\rho_- + \rho_+}{2}$)
    aller en 10 :
fin si

TAB. 4.4 – Algorithme de Wolfe dans le cadre d'une recherche de pas pour une méthode de descente.

#### 4.2.3.2 Règle de Wolfe (1969)

On choisit deux nombres $m_1$ et $m_2$, avec $0 < m_1 < m_2 < 1$ (par exemple $m_1 = 0.1$ et $m_2 = 0.7$), et on recherche $\rho$ qui vérifie

$$\begin{cases} q(\rho) \leq q(0) + m_1 \rho q'(0) \\ q(\rho) \geq m_2 q'(0) \end{cases}$$

#### 4.2.3.3 Mise en oeuvre des règles précédentes dans un algorithme général utilisant des directions de descente

Soit $J$ la fonction à minimiser. A l'itération $k$, nous avons $\mathbf{x} = \mathbf{x}^{(k)}$ et $\mathbf{d} = \mathbf{d}^{(k)}$, et on veut calculer $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho \mathbf{d}^{(k)}$ pour une valeur $\rho$ à déterminer. L'algorithme associé à la règle de Wolfe est alors donné par la table 4.4, celui pour la méthode de Goldstein étant similaire.

## 4.3 Quelques notions sur les algorithmes

Intéressons nous maintenant au développement d'algorithmes numériques de résolution des problèmes de minimisation destinés à être mis en oeuvre sur calculateurs. Nous ne verrons ici que des algorithmes de base, l'optimisation étant un vaste domaine de recherches et d'applications. Nous ne nous intéresserons ici qu'à l'optimisation locale, la recherche d'un extremum global étant hors de portée de cette introduction. Par

ailleurs, l'hypothèse de différentiabilité nous suivra tout le long de cet exposé ; encore une fois, l'optimisation non différentiable n'est pas traitée ici.

Un algorithme... Mais qu'est qu'un algorithme ?

**Définition 16** *Un algorithme itératif est défini par une application vectorielle* $\mathbb{A} : \mathbb{R}^n \to \mathbb{R}^n$ *qui génère une suite de champs de vecteurs* $(\mathbf{x}^{(k)})_{k\geq 0}$ *par une construction typiquement de la forme*

```
choisir x^(0) (phase d'initialisation de l'algorithme)
calculer x^(k+1) = A(x^(k)) (k-ième itération))
```

Bien sûr, ce que nous espérons, c'est que notre suite $(\mathbf{x}^{(k)})_{k\geq 0}$ converge vers une limite $\mathbf{x}^*$ qui sera effectivement notre point de minimum relatif. On dit que l'algorithme converge vers la solution du problème de minimisation si c'est la cas. Lorsque l'on a un algorithme donné, deux mesures importantes de son efficacité sont : d'une part la vitesse de convergence, d'autre part, sa complexité calculatoire. La vitesse de convergence mesure "la rapidité" avec laquelle la suite $(\mathbf{x}^{(k)})_{k\geq 0}$ converge vers le point $\mathbf{x}^*$. La complexité mesure le coût des opérations nécessaires pour obtenir une itération, le coût global étant le coût d'une itération multiplié par le nombre d'itérations pour obtenir la solution escomptée avec une certaine précision $\varepsilon$ fixée *a priori*. On prend généralement les appellations suivantes. On introduit l'erreur vectorielle sur la solution : $\mathbf{e}^{(k)} = \mathbf{x}^* - \mathbf{x}^{(k)}$. Si sa norme (euclidienne) $e^{(k)} = \left\|\mathbf{e}^{(k)}\right\|$ décroît linéairement, alors, on dit que la vitesse de convergence est linéaire. Plus mathématiquement, cette propriété s'exprime par une relation du type

$$(\exists C \in [0;1[)(\exists k_0 \in \mathbb{N})(\forall k \geq k_0)(e^{(k+1)} \leq Ce^{(k)})$$

On voit bien à ce niveau que la vitesse de convergence est un notion asymptotique. Elle n'est pas nécessairement observable à la première itération. Si nous observons une relation du type $e^{(k+1)} \leq \gamma^{(k)} e^{(k)}$, nous dirons que la vitesse est superlinéaire si $\lim_{k\to+\infty} \gamma^{(k)} = 0$, pour $\gamma^{(k)} \geq 0$, pour tout $k \geq 0$. On parle de convergence géométrique lorsque la suite $(\gamma^{(k)})_k$ est une suite géométrique. La méthode est dite d'ordre $p$ si l'on a une relation du type

$$(\exists C \in [0;1[)(\exists k_0 \in \mathbb{N})(\forall k \geq k_0)(e^{(k+1)} \leq C(e^{(k)})^p)$$

Si $p = 2$, nous dirons que la vitesse de convergence est quadratique. Finalement, si la convergence a lieu seulement pour des $\mathbf{x}^{(0)}$ voisins de $\mathbf{x}^*$, nous parlerons de convergence locale, sinon, nous dirons globale.

## 4.4 Méthodes de gradient

La méthode du gradient fait partie des classes de méthodes dites de descente. Quelle est l'idée cachée derrière ces méthodes ? Considérons un point de départ $\mathbf{x}^{(0)}$ et cherchons à minimiser une fonction $J$. Puisque l'on veut atteindre $\mathbf{x}^*$, nous cherchons à avoir : $J(\mathbf{x}^{(1)}) < J(\mathbf{x}^{(0)})$. Une forme particulièrement simple est de chercher $\mathbf{x}^{(1)}$ tel que le vecteur $\mathbf{x}^{(1)} - \mathbf{x}^{(0)}$ soit colinéaire à une direction de descente $\mathbf{d}^{(0)} \neq \mathbf{0}$. Nous le noterons : $\mathbf{x}^{(1)} - \mathbf{x}^{(0)} = \rho^{(0)}\mathbf{d}^{(1)}$, où $\rho^{(0)}$ est le pas de descente de la méthode. On peut alors itérer de cette manière en

poser $k = 0$
choisir $\mathbf{x}^{(0)}$
tant que $(\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| \geq \varepsilon)$ et $(k \leq k^{\max})$ faire
  calculer $\mathbf{d}^{(k)} = -\nabla J(\mathbf{x}^{(k)})$
  calculer $\rho^{(k)}$
  poser $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}$
fin tant que

TAB. 4.5 – Algorithme du gradient.

se donnant $\mathbf{x}^{(k)}$, $\mathbf{d}^{(k)}$ et $\rho^{(k)}$ pour atteindre $\mathbf{x}^{(k+1)}$ par

$$\begin{aligned} &\texttt{choisir } \mathbf{x}^{(0)} \\ &\texttt{calculer } \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)} \end{aligned}$$

avec $\mathbf{d}^{(k)} \in \mathbb{R}^{n*}$ et $\rho^{(k)} > 0$. De nombreux choix existent pour $\mathbf{d}^{(k)}$ et $\rho^{(k)}$.

La première question consiste à choisir la direction de descente. Rappelons que le développement de Taylor de $J$ au premier ordre donne au voisinage de $\mathbf{x}^{(k+1)}$

$$J(\mathbf{x}^{(k+1)}) = J(\mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}) = J(\mathbf{x}^{(k)}) + \rho^{(k)}\nabla J(\mathbf{x}^{(k)}) \cdot \mathbf{d}^{(k)} + \rho^{(k)})\|\mathbf{d}^{(k)}\|E(\mathbf{x}^{(k)}; \rho^{(k)}\mathbf{d}^{(k)}),$$

où $\lim_{\rho^{(k)}\mathbf{d}^{(k)} \to 0} E(\mathbf{x}^{(k)}; \rho^{(k)}\mathbf{d}^{(k)}) = 0$. Or, puisque l'on désire avoir : $J(\mathbf{x}^{(1)}) < J(\mathbf{x}^{(0)})$, une solution évidente consiste à prendre

$$\mathbf{d}^{(k)} = -\nabla J(\mathbf{x}^{(k)}),$$

puisqu'àlors

$$J(\mathbf{x}^{(k+1)}) - J(\mathbf{x}^{(k)}) = -\rho^{(k)} \left\|\nabla J(\mathbf{x}^{(k)})\right\|^2 + o(\rho^{(k)}).$$

Nous voyons que si $\rho^{(k)}$ est suffisamment petit, $x^{(k+1)}$ minimisera mieux $J$ que ne le faisait $x^{(k)}$. La méthode obtenue avec le choix $\mathbf{d}^{(k)} = -\nabla J(\mathbf{x}^{(k)})$ est appelée méthode du gradient. Lorsque l'on travaille sur une résolution numérique d'un problème, on se donne en général un critère d'arrêt de la forme : $\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| < \varepsilon$. De plus, puisque la convergence n'est pas toujours assurée, une règle de base est de fixer un nombre maximum d'itérations $k^{\max}$. On obtient alors l'algorithme présenté table 4.5 et dit du gradient.

Même si ces méthodes sont conceptuellement très simples et qu'elles peuvent être programmées directement, elles sont souvent lentes dans la pratique. Elles convergent mais sous des conditions de convergence souvent complexes. A titre d'exemple, donnons le résultat suivant.

**Théorème 21** *Soit $J$ une fonction de classe $\mathcal{C}^1$ de $\mathbb{R}^n$ dans $\mathbb{R}$, $\mathbf{x}^*$ un minimum de $J$. Supposons que*

*i) $J$ est $\alpha$-elliptique, c'est-à-dire,*

$$(\exists \alpha > 0)(\forall (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n)(\nabla J(\mathbf{x}) - \nabla J(\mathbf{y})) \cdot (\mathbf{x} - \mathbf{y}) \geq \alpha \left\|\mathbf{x} - \mathbf{y}\right\|^2.$$

*ii) l'application $\nabla J$ est lipschitzienne*

$$(\exists M > 0)(\forall (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n)(\|\nabla J\mathbf{x} - \nabla J\mathbf{y}\| \leq M \|\mathbf{x} - \mathbf{y}\|).$$

*S'il existe deux réels $a$ et $b$ tels que $\rho^{(k)}$ satisfasse $0 < a < \rho^{(k)} < b < 2\alpha$, pour tout $k \geq 0$, alors, la méthode du gradient définie par*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \rho^{(k)} \nabla J(\mathbf{x}^{(k)})$$

*converge pour tout choix de $\mathbf{x}^{(0)}$ de façon géométrique*

$$\exists \beta \in ]0;1[, \left\| \mathbf{x}^{(k+1)} - \mathbf{x}^* \right\| \leq \beta^k \left\| \mathbf{x}^{(0)} - \mathbf{x}^* \right\|.$$

Le choix du pas $\rho^{(k)}$ peut être effectué de la manière suivante

- soit $\rho^{(k)} = \rho$ est fixé *a priori* : c'est ce que l'on appelle la méthode du gradient à pas fixe ou constant,
- soit $\rho^{(k)}$ est choisi comme le minimum de la fonction $q(\rho) = J(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)}))$ : c'est ce que l'on appelle la méthode du gradient à pas optimal,
- ou, soit $\rho^{(k)}$ est calculé par les méthodes présentées précédemment.

Dans le cas du gradient à pas optimal, nous avons le même résultat de convergence que précédemment sous des hypothèses faibles sur $J$.

**Théorème 22** *Soit $J$ une fonction de classe $\mathcal{C}^1$ de $\mathbb{R}^n$ dans $\mathbb{R}$, $\mathbf{x}^*$ un minimum de $J$. On suppose que $J$ est $\alpha$-elliptique. Alors, la méthode du gradient à pas optimal converge pour tout choix du vecteur d'initialisation $\mathbf{x}^{(0)}$.*

**Remarque 4** *Même pour le gradient à pas optimal qui est en principe la meilleure de ces méthodes d'un point de vue de la rapidité de convergence, celle-ci peut être lente car altérée par un mauvais conditionnement de la matrice hessienne de $J$. Par ailleurs, on peut considérer des critères de convergence sur le gradient de $J$ en $\mathbf{x}^{(k)}$ : $\left\|\nabla J(\mathbf{x}^{(k)})\right\| < \varepsilon_1$*

## 4.5 Méthode du gradient conjugué

Considérons une matrice $A$, définie positive, et soit $J$ la fonctionnelle quadratique définie par

$$\begin{aligned} J : \mathbb{R}^n &\to \mathbb{R} \\ \mathbf{x} &\mapsto J(\mathbf{x}) = \frac{1}{2} A\mathbf{x} \cdot \mathbf{x} - \mathbf{b} \cdot \mathbf{x}. \end{aligned}$$

La fonction $J$ est alors une fonctionnelle strictement convexe (à montrer) deux fois continûment différentiable. Un calcul de son gradient donne : $\nabla J(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$. Par conséquent, le minimum (unique et global) de $J$ est réalisé en $\mathbf{x}^*$ tel que : $A\mathbf{x}^* = \mathbf{b}$.

Afin de développer le gradient conjugué, introduisons la notion de direction conjuguée.

**Définition 17** *Nous dirons que deux vecteurs (ou directions)* $\mathbf{d}_1$ *et* $\mathbf{d}_2$ *sont conjugués pour la matrice* $A$ *si :* $A\mathbf{d}_2 \cdot \mathbf{d}_1 = 0$.

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

$$(\mathbf{x}, \mathbf{y})_A = A\mathbf{x} \cdot \mathbf{y}, \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n.$$

Faisons l'hypothèse que nous connaissons $k$ directions conjuguées $\mathbf{d}^{(0)}, ..., \mathbf{d}^{(k-1)}$. La méthode de descente consiste, en partant d'un point $\mathbf{x}^{(0)} \in \mathbb{R}^n$, à calculer par des itérations successives $\mathbf{x}^{(k+1)}$ tel qu'il satisfasse

$$J(\mathbf{x}^{(k+1)}) = J(\mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}) = \min_{\rho \in \mathbb{R}} J(\mathbf{x}^{(k)} + \rho \mathbf{d}^{(k)}).$$

On peut expliciter la valeur de $\rho^{(k)}$ en utilisant la condition de minimum au premier ordre, à savoir

$$\nabla J(\mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}) \cdot \mathbf{d}^{(k)} = 0,$$

et plus explicitement dans notre cas

$$\rho^{(k)} = -\frac{(A\mathbf{x}^{(k)} - \mathbf{b}) \cdot \mathbf{d}^{(k)}}{\left\|\mathbf{d}^{(k)}\right\|_A^2} = -\frac{(\mathbf{x}^{(k)}, \mathbf{d}^{(k)})_A}{\left\|\mathbf{d}^{(k)}\right\|_A^2} + \frac{\mathbf{b} \cdot \mathbf{d}^{(k)}}{\left\|\mathbf{d}^{(k)}\right\|_A^2} \tag{4.2}$$

en posant $\left\|\mathbf{d}^{(k)}\right\|_A = (\mathbf{d}^{(k)}, \mathbf{d}^{(k)})_A^{1/2}$. Or, par définition, $\mathbf{x}^{(k)} - \mathbf{x}^{(0)}$ peut s'exprimer selon les vecteurs $\mathbf{d}^{(0)}, ..., \mathbf{d}^{(k-1)}$ puisque

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \rho^{(k-1)}\mathbf{d}^{(k-1)} = \mathbf{x}^{(k-2)} + \rho^{(k-2)}\mathbf{d}^{(k-2)} + \rho^{(k-1)}\mathbf{d}^{(k-1)} = ... = \mathbf{x}^{(0)} + \sum_{\ell=0}^{k-1} \rho^{(\ell)}\mathbf{d}^{(\ell)}$$

Ainsi, nous pouvons simplifier (4.2) par conjugaison pour écrire

$$\rho^{(k)} = -\frac{(\mathbf{x}^{(0)}, \mathbf{d}^{(k)})_A}{\left\|\mathbf{d}^{(k)}\right\|_A^2} + \frac{\mathbf{b} \cdot \mathbf{d}^{(k)}}{\left\|\mathbf{d}^{(k)}\right\|_A^2} = -\frac{(\mathbf{r}^{(0)}, \mathbf{d}^{(k)})_A}{\left\|\mathbf{d}^{(k)}\right\|_A^2},$$

où le vecteur résidu $\mathbf{r}^{(0)}$ à l'instant initial est défini par : $\mathbf{r}^{(0)} = A\mathbf{x}^{(0)} - \mathbf{b}$.

Le succès de l'algorithme du gradient conjugué est intimement lié à la proposition importante suivante.

**Proposition 9** *Le point* $\mathbf{x}^{(k)}$ *est le minimum de* $J$ *sur le sous-espace affine passant par* $\mathbf{x}^{(0)}$ *engendré par les vecteurs* $\{\mathbf{d}^{(0)}, ..., \mathbf{d}^{(k-1)}\}$.

Une conséquence fondamentale de la proposition précédente est que, si l'on est capable de trouver $n$ directions conjuguées $\{\mathbf{d}^{(0)}, ..., \mathbf{d}^{(n-1)}\}$, on a résolu le problème de minimisation puisque $\{\mathbf{d}^{(0)}, ..., \mathbf{d}^{(n-1)}\}$ est une base de $\mathbb{R}^n$. En fait, l'algorithme du gradient conjugué consiste à construire simultanément ces directions conjuguées par le procédé de Gram-Schmidt. Plus précisément, l'algorithme est décrit dans la table 4.6.

$k = 0$
choisir $\mathbf{x}^{(0)} \in \mathbb{R}^n$
choisir $\varepsilon > 0$
choisir $\varepsilon_1 > 0$
poser $\mathbf{r}^{(0)} = \nabla J(\mathbf{x}^{(0)})$
tant que $\left(\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| \geq \varepsilon\right)$ et $(k \leq k^{\max})$ faire
    si $\left(\left\|\mathbf{r}^{(k)}\right\| < \varepsilon_1\right)$ alors arrêt
    sinon
        si $(k = 0)$ alors
            poser $\mathbf{d}^{(k)} = \mathbf{r}^{(k)}$
        sinon
            calculer $\alpha^{(k)} = -\dfrac{(\mathbf{r}^{(k)}, \mathbf{d}^{(k-1)})_A}{\left\|\mathbf{d}^{(k-1)}\right\|_A^2}$
            poser $\mathbf{d}^{(k)} = \mathbf{r}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k-1)}$
        fin si
        calculer $\rho^{(k)} = -\dfrac{(\mathbf{r}^{(k)}, \mathbf{d}^{(k)})}{\left\|\mathbf{d}^{(k)}\right\|_A^2}$
        poser $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}$
        calculer $\mathbf{r}^{(k+1)} = A\mathbf{x}^{(k+1)} - \mathbf{b}$
        poser $k = k + 1$
    fin si
fin tant que

TAB. 4.6 – Algorithme du gradient conjugué pour une fonctionnelle quadratique.

$k = 0$
choisir $\mathbf{x}^{(0)} \in \mathbb{R}^n$
choisir $\varepsilon > 0$
choisir $\varepsilon_1 > 0$
poser $\mathbf{r}^{(0)} = \nabla J(\mathbf{x}^{(0)})$
tant que $\left(\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| \geq \varepsilon\right)$ et $(k \leq k^{\max})$ faire
    si $\left(\left\|\mathbf{r}^{(k)}\right\| < \varepsilon_1\right)$ alors arrêt
    sinon
        si $(k = 0)$ alors
            poser $\mathbf{d}^{(k)} = \mathbf{r}^{(k)}$
        sinon
            calculer $\alpha^{(k)} = \dfrac{\left\|\mathbf{r}^{(k)}\right\|^2}{\left\|\mathbf{r}^{(k-1)}\right\|^2}$
            poser $\mathbf{d}^{(k)} = \mathbf{r}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k-1)}$
        fin si
        si $(\mathbf{d}^{(k)} \cdot \mathbf{r}^{(k)} \geq 0)$ alors
            poser $\mathbf{d}^{(k)} = \mathbf{r}^{(k)}$
        sinon
            rechercher un pas $\rho^{(k)}$ approchant le minimum de $J(\mathbf{x}^{(k)} + \rho\mathbf{d}^{(k)})$ par
            $\nabla J(\mathbf{x}^{(k)} + \rho\mathbf{d}^{(k)}) \cdot \mathbf{d}^{(k)} = 0$
        fin si
        poser $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}$
        poser $\mathbf{r}^{(k+1)} = \nabla J(\mathbf{x}^{(k+1)})$
        poser $k = k + 1$
    fin si
fin tant que

TAB. 4.7 – Algorithme du gradient conjugué pour une fonctionnelle générale.

L'algorithme de Gram-Schmidt peut, dans certains cas, s'avérer instable. Dûe aux erreurs d'arrondis, la méthode peut mettre un peu plus que $n$ itérations pour converger.

Rappelons que l'algorithme présenté ici était particularisé au cas d'une fonctionnelle quadratique. On peut étendre l'algorithme pour une fonctionnelle $J$ quelconque de manière efficace comme nous le détaillons dans la table 4.7.

## 4.6 Les méthodes de Newton et quasi-Newton

La méthode de Newton n'est pas a proprement parlé une méthode d'optimisation. C'est une méthode de recherche de zéros d'une fonction $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$ selon : $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. L'idée de cette méthode ici est de résoudre l'équation $\nabla J(\mathbf{x}) = \mathbf{0}$, condition nécessaire de premier ordre pour la détection d'extrema d'une fonction. L'équation $\nabla J(\mathbf{x}) = \mathbf{0}$ est donnée par un système $n \times n$ d'équations non linéaires. La méthode s'écrit

poser $k = 0$
choisir $x^{(0)}$ dans un voisinage de $x^*$
choisir $\varepsilon > 0$
tant que $(|x^{(k+1)} - x^{(k)}| \geq \varepsilon)$ et $(k \leq k^{\max})$ faire
    poser $x^{(k+1)} = x^{(k)} - \dfrac{f(x^{(k)})}{f'(x^{(k)})}$
    poser $k = k + 1$
fin tant que

TAB. 4.8 – Méthode de Newton unidimensionnelle.

formellement

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [D^2 J(\mathbf{x}^{(k)})]^{-1} \nabla J(\mathbf{x}^{(k)}).$$

### 4.6.1 Méthodes de Newton

Cherchons à résoudre, pour $f : \mathbb{R} \to \mathbb{R}$, une équation $f(x) = 0$. Nous supposons que $f$ est de classe $\mathcal{C}^1$. L'algorithme de Newton est donné table 4.8.

Une interpétation géométrique simple de cette méthode est donnée à partir de la tangente au point $x^{(k)}$. La convergence de la méthode doit être précisée ainsi que la définition de la suite $(f'(x^{(k)}))_k$.

Généralisons maintenant cette méthode au cas d'un champ scalaire $J$ donné de $\mathbb{R}^n$ dans $\mathbb{R}$. Soit $\mathbf{F}$ une fonction de classe $\mathcal{C}^1$. On suppose que l'équation $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ possède au moins une solution notée $\mathbf{x}^*$ et que la matrice $D\mathbf{F}(\mathbf{x}^*)$ est une matrice inversible. La continuité de $D\mathbf{F}$ permet en fait d'assurer l'inversibilité de $D\mathbf{F}(\mathbf{x}^{(k)})$ pour tout point $\mathbf{x}^{(k)}$ se trouvant dans un voisinage de $\mathbf{x}^*$ et permet de définir l'itéré $\mathbf{x}^{(k+1)}$. L'extension de la seconde étape est réalisée par la résolution du système linéaire

$$[D\mathbf{F}(\mathbf{x}^{(k)})]\boldsymbol{\delta}^{(k)} = \mathbf{F}(\mathbf{x}^{(k)}),$$

puis on pose $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \boldsymbol{\delta}^{(k)}$. En résumé, l'algorithme de Newton peut s'écrire sous la forme présentée table 4.9.

En ce qui concerne la convergence de ce dernier algorithme, nous avons le résultat suivant.

**Théorème 23** *Soit $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$ une fonction de classe $\mathcal{C}^1$ et $\mathbf{x}^*$ un zéro de $\mathbf{F}$. On suppose que ce zéro est isolé et que $D\mathbf{F}(\mathbf{x}^*)$ est inversible ($D\mathbf{F}$ désigne la dérivée première de $\mathbf{F}$). Alors, il existe une boule $\mathcal{B}(\mathbf{x}^*)$ telle que, pour tout point $\mathbf{x}^{(0)} \in \mathcal{B}(\mathbf{x}^*)$, la suite $(\mathbf{x}^{(k)})_k$ définie par la méthode de Newton est entièrement contenue dans $\mathcal{B}$ et converge vers $\mathbf{x}^*$, seul zéro de $\mathbf{F}$ dans $\mathcal{B}$. De plus, la convergence est géométrique : il existe $\beta \in ]0;1[$ tel que*

$$\forall k \geq 0, \quad \left\| \mathbf{x}^{(k+1)} - \mathbf{x}^* \right\| \leq \beta^k \left\| \mathbf{x}^{(0)} - \mathbf{x}^* \right\|.$$

*Par conséquent, si nous choisissons $\mathbf{x}^{(0)}$ "suffisamment près" de $\mathbf{x}^*$, la méthode de Newton converge.*

```
poser k = 0
choisir x^(0) dans un voisinage de x*
choisir ε > 0
tant que (‖x^(k+1) − x^(k)‖ ≥ ε) et (k ≤ k^max) faire
      resoudre le systeme lineaire [DF(x^(k))]δ^(k) = F(x^(k))
      poser x^(k+1) = x^(k) − δ^(k)
      poser k = k + 1
fin tant que
```

TAB. 4.9 – Méthode de Newton multidimensionnelle.

Un des gros problèmes de cette méthode est que *le choix de* $\mathbf{x}^{(0)}$ *joue un grand rôle sur la convergence ou non de la méthode de Newton*. La méthode est très sensible à l'initialisation. Il peut aussi arriver que la méthode converge vers un extremum qui n'est pas celui cherché. Une approche possible consiste à faire tourner quelques itérations d'une méthode de gradient pour approcher $\mathbf{x}^*$ et de considérer l'itéré résultant comme le point de départ de la méthode de Newton. L'avantage de la méthode de Newton est sa grande rapidité de convergence : *la convergence de la méthode de Newton est quadratique*. Un des inconvénients de la méthode de Newton réside également dans le fait que nous avons à évaluer et "à inverser" (en fait résoudre un système plein associé) la matrice $D\mathbf{F}(\mathbf{x}^{(k)})$ à chaque itération. Certaines méthodes proposent d'utiliser non pas $D\mathbf{F}(\mathbf{x}^{(k)})$ comme matrice mais plutôt une approximation de celle-ci (plus précisemment, dans les méthodes de quasi-Newton décrites ci-dessous, on construit une suite de matrices $S^{(k)}$ qui approchent $[D\mathbf{F}(\mathbf{x}^{(k)})]^{-1}$).
Lorsque l'on utilise la méthode de Newton pour résoudre : $\nabla J(\mathbf{x}^*) = \mathbf{0}$, on prend bien sûr $\mathbf{F} = \nabla J$. La méthode donnera alors les points critiques de $J$, la propriété de minimum étant à vérifier *a posteriori*. Dans ce cas, $D\mathbf{F}$ est la matrice hessienne $D^2J$.

### 4.6.2 Méthode de quasi-Newton de Lenvenberg-Marquardt (avec recherche linéaire)

Nous donnons table 4.10 un algorithme de type quasi-Newton (avec recherche linéaire d'Armijo) qui se trouve être à convergence quadratique sous des hypothèses standards (remarquer que l'hypothèse de localité du point de départ n'est plus demandée).

### 4.6.3 Méthode de quasi-Newton DFP et BFGS

L'idée ici est d'imiter l'algorithme de Newton tout en évitant de calculer $D^2J$ et "son inverse". Plus précisément, cela signifie que, à l'itération $k$, nous allons chercher à construire une approximation $S^{(k)}$ symétrique définie positive de $[D^2J(\mathbf{x}^{(k)})]^{-1}$ et $\rho^{(k)}$ un paramètre positif fourni par un algorithme de minimisation unidimensionnel le long de la direction $\mathbf{d}^{(k)} = -S^{(k)}\nabla J(\mathbf{x}^{(k)})$ tels que l'opération classique

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [D^2J(\mathbf{x}^{(k)})]^{-1}\nabla J(\mathbf{x}^{(k)}),$$

choisir $(\alpha, \beta, \gamma) \in ]0;1[\times]0;1[\times]0;1[$
poser $k = 0$
choisir $\mathbf{x}^{(0)} \in \mathbb{R}^n$
calculer $\mu^{(0)} = \left\|\nabla J(\mathbf{x}^{(0)})\right\|^2$
choisir $\varepsilon > 0$
tant que $(\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| \geq \varepsilon)$ et $(k \leq k^{\max})$ faire
    resoudre le systeme lineaire
        $[\mu^{(k)} Id + D^2 J(\mathbf{x}^{(k)})^T D^2 J(\mathbf{x}^{(k)})]\mathbf{d}^{(k)} = -D^2 J(\mathbf{x}^{(k)}) \nabla J(\mathbf{x}^{(k)})$
    si $(\left\|\nabla J(\mathbf{x}^{(k)} + \mathbf{d}^{(k)})\right\| \leq \gamma \left\|\nabla J(\mathbf{x}^{(k)})\right\|)$ alors
        poser $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{d}^{(k)}$
        aller en 4
    sinon
        aller en 3
    fin si
    debut 3 :
        soit $m_k$ le plus petit entier positif tel que
        $\left\|\Psi(\mathbf{x}^{(k)} + \beta^m \mathbf{d}^{(k)})\right\|^2 - \left\|\Phi(\mathbf{x}^{(k)})\right\|^2 \leq \alpha\beta^m \mathbf{d}^{(k)} \cdot \nabla\Psi(\mathbf{x}^{(k)})$
        avec $\Psi(\mathbf{x}) = \frac{1}{2}\left\|\nabla(\mathbf{x})\right\|^2$
        poser $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \beta^m \mathbf{d}^{(k)}$
        aller en 4
    fin 3 :
    debut 4 :
        calculer $\mu^{(k+1)} = \left\|\nabla J(\mathbf{x}^{(k+1)})\right\|^2$
        poser $k = k + 1$
    fin 4 :
fin tant que

TAB. 4.10 – Méthode de quasi-Newton de Lenvenberg-Marquardt (avec recherche linéaire).

choisir $S^{(0)}$, matrice symetrique definie positive
poser $k = 0$
choisir $\mathbf{x}^{(0)} \in \mathbb{R}^n$
choisir $\varepsilon > 0$
tant que $(\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| \geq \varepsilon)$ et $(k \leq k^{\max})$ faire
  calculer $\rho^{(k)}$ par une methode de recherche lineaire sur $\mathbf{d}^{(k)} = -S^{(k)}\nabla J(\mathbf{x}^{(k)})$
  poser $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}$
  poser $\boldsymbol{\delta}^{(k+1)} = \rho^{(k)}\mathbf{d}^{(k)}$
  calculer $\boldsymbol{\gamma}^{(k)} = \nabla J(\mathbf{x}^{(k+1)}) - \nabla J(\mathbf{x}^{(k)})$
  calculer $S^{(k+1)} = S^{(k)} + \dfrac{\boldsymbol{\delta}^{(k)}\boldsymbol{\delta}^{(k)T}}{\boldsymbol{\delta}^{(k)T}\boldsymbol{\gamma}^{(k)}} - \dfrac{\boldsymbol{\delta}^{(k)}\boldsymbol{\gamma}^{(k)}\boldsymbol{\gamma}^{(k)T}\boldsymbol{\delta}^{(k)}}{\boldsymbol{\gamma}^{(k)T}\boldsymbol{\delta}^{(k)}\boldsymbol{\gamma}^{(k)}}$
fin tant que

TAB. 4.11 – Algorithme de Davidson-Fletcher-Powell (DFP).

soit remplacée par l'opération plus simple et beaucoup moins coûteuse

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \rho^{(k)} S^{(k)} \nabla J(\mathbf{x}^{(k)}).$$

Le but est de calculer "une bonne approximation" $S^{(k)}$ de $[D^2 J(\mathbf{x}^{(k)})]^{-1}$, c'est-à-dire telle que la différence soit petite pour une norme matricielle.

#### 4.6.3.1 Algorithme DFP (Davidson-Fletcher-Powell)

Cette méthode permet notamment, pour une fonctionnelle quadratique, de construire l'inverse du hessien. Elle engendre également des directions conjuguées. L'algorithme est donné table 4.11 (pour une matrice $S$ donnée, $S^T$ désigne la matrice transposée).

Les bonnes propriétés de cet algorithme sont résumées dans le théorème suivant.

**Théorème 24** *A l'étape $k$, si $S^{(k)}$ est symétrique définie positive et si la recherche linéaire est exacte (ou bien si $\boldsymbol{\delta}^{(k)T}\boldsymbol{\gamma}^{(k)} > 0$), alors la matrice $S^{(k+1)}$ est symétrique définie positive. Si $J$ est quadratique, de hessien $D^2 J$, alors la méthode DFP est telle que*

$$\boldsymbol{\delta}^{(i)T} D^2 J \boldsymbol{\delta}^{(j)} = 0, \quad 0 \leq i \leq j \leq k,$$
$$\boldsymbol{\delta}^{(k+1)} D^2 J \boldsymbol{\delta}^{(i)} = \boldsymbol{\delta}^{(i)}, \quad 0 \leq i \leq k.$$

Dans le cas quadratique, ceci implique que la méthode converge en $n$ itérations et $S^{(n)} = [D^2 J]^{-1}$.

**Remarque 5** *Si $S^{(0)} = Id$ ($Id$ est la matrice identité ici), on a la méthode du gradient conjugué. Cette méthode est sensible à la précision de la recherche linéaire.*

choisir $S^{(0)}$, matrice symetrique definie positive
poser $k = 0$
choisir $\mathbf{x}^{(0)} \in \mathbb{R}^n$
choisir $\varepsilon > 0$
tant que $(\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| \geq \varepsilon)$ et $(k \leq k^{\max})$ faire
    calculer $\rho^{(k)}$ par une methode de recherche lineaire sur $\mathbf{d}^{(k)} = -S^{(k)}\nabla J(\mathbf{x}^{(k)})$
    poser $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}$
    poser $\boldsymbol{\delta}^{(k+1)} = \rho^{(k)}\mathbf{d}^{(k)}$
    calculer $\boldsymbol{\gamma}^{(k)} = \nabla J(\mathbf{x}^{(k+1)}) - \nabla J(\mathbf{x}^{(k)})$
    calculer $S^{(k+1)} = S^{(k)} + (1 + \dfrac{\boldsymbol{\gamma}^{(k)T}\boldsymbol{\delta}^{(k)}\boldsymbol{\gamma}^{(k)}}{\boldsymbol{\delta}^{(k)T}\boldsymbol{\gamma}^{(k)}})\dfrac{\boldsymbol{\delta}^{(k)}\boldsymbol{\delta}^{(k)T}}{\boldsymbol{\delta}^{(k)T}\boldsymbol{\gamma}^{(k)}} - \dfrac{\boldsymbol{\delta}^{(k)}\boldsymbol{\gamma}^{(k)T}\boldsymbol{\delta}^{(k)} + \boldsymbol{\delta}^{(k)}\boldsymbol{\gamma}^{(k)}\boldsymbol{\delta}^{(k)T}}{\boldsymbol{\delta}^{(k)T}\boldsymbol{\gamma}^{(k)}}$
fin tant que

TAB. 4.12 – Algorithme de Broyden-Fletcher-Goldfarb-Shanno (BFGS).

#### 4.6.3.2 Méthode de Broyden-Fletcher-Goldfarb-Shanno (BFGS)

Cette dernière méthode est considérée comme très robuste et est moins sensible aux erreurs dans les recherches linéaires (cf. table 4.12).

Nous pourrions aussi, dans le cas quadratique, citer un théorème de convergence. La robustesse des précédents algorithmes est intimement liée à la recherche du pas optimal $\rho^{(k)}$. En pratique, on utilise souvent la règle d'Armijo.

## 4.7 Quelques remarques

Il existe des algorithmes permettant des recherches d'extrema pour des fonctions moins régulières, c'est ce que l'on appelle l'analyse non différentiable. Ceci rejoint les méthodes liées à la notion de sous-différentiel. Il existe également des algorithmes stochastiques (recuit simulé par exemple).

## 4.8 Exercices

**Exercice 4.1** (Méthode du gradient à pas constant).

On veut minimiser la fonctionnelle $J(x) = \frac{1}{2}(Ax, x) - (b, x)$, où $\mathbf{b} \in \mathbb{R}^n$ et $A$ est une matrice réelle symétrique définie positive. Cela revient à résoudre le système $A\mathbf{x} = \mathbf{b}$. Pour cela, nous allons employer la méthode du gradient à pas constant. Soit $\mathbf{x}^*$ la solution de ce système. On propose l'algorithme suivant : on se donne un vecteur initial $\mathbf{x}^{(0)}$, et on calcule la suite d'itérés par l'algorithme

$$\begin{aligned} \mathbf{r}^{(k)} &= \mathbf{b} - A\mathbf{x}^{(k)} \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha\mathbf{r}^{(k)} \end{aligned} \tag{4.3}$$

où $\alpha$ est une constante réelle donnée.

1. Soit $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$, pour $k \geq 0$. Donner une relation entre $\mathbf{e}^{(k)}$ et $\mathbf{e}^{(0)}$.
2. Montrer que l'agorithme converge si et seulement si

$$0 < \alpha < \frac{2}{\lambda_n},$$

où $\lambda_n$ est la plus grande valeur propre de $A$.

3. Donner le meilleur choix pour $\alpha$ en fonction des valeurs propres de $A$. Que concluez-vous ?

**Exercice 4.2** (Méthode du gradient à pas optimal).

Soit $A \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive et $\mathbf{b}$ un vecteur de $\mathbb{R}^n$. On note $\lambda_i$, $1 \leq i \leq n$, les valeurs propres de $A$ rangées par ordre croissant :

$$0 < \lambda_1 \leq ... \leq \lambda_n.$$

On définit la fonctionnelle quadratique $J$ par

$$J(\mathbf{x}) = \frac{1}{2} A\mathbf{x} \cdot \mathbf{x} - \mathbf{b} \cdot \mathbf{x},$$

où $\cdot$ désigne le produit scalaire usuel de $\mathbb{R}^n$.

On considère la méthode itérative suivante pour résoudre du système linéaire $A\mathbf{x} = \mathbf{b}$ : on se donne un point initial $\mathbf{x}^{(0)}$ de $\mathbb{R}^n$, on définit le résidu initial $\mathbf{r}^{(0)} = A\mathbf{x}^{(0)} - \mathbf{b}$, et tant que le résidu $\mathbf{r}^{(k)}$ à l'ordre $k$ est non nul, on pose

$$\begin{aligned} \alpha^{(k)} &= \frac{\mathbf{r}^{(k)} \cdot \mathbf{r}^{(k)}}{A\mathbf{r}^{(k)} \cdot \mathbf{r}^{(k)}} \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{r}^{(k)} \\ \mathbf{r}^{(k+1)} &= \mathbf{b} - A\mathbf{x}^{(k+1)} \end{aligned} \tag{4.4}$$

Bien sûr, si $\mathbf{r}^{(k)} = 0$, alors $\mathbf{x}^{(k)}$ est solution du système linéaire.

Dans tout l'exercice, nous ferons l'hypothèse que $\mathbf{r}^{(k)} \neq \mathbf{0}$.

1. Montrer que $J$ est différentiable et déterminer sa différentielle.
2. Montrer que $\alpha^{(k)}$ est l'unique réel qui minimise sur $\mathbb{R}$ la fonction $\alpha \to J(\mathbf{x}^{(k)} + \alpha \mathbf{r}^{(k)})$.
3. Evaluer la différence : $J(\mathbf{x}^{(k+1)}) - J(\mathbf{x}^{(k)})$.
4. Montrer que l'on a

$$A^{-1}\mathbf{r}^{(k+1)} \cdot \mathbf{r}^{(k+1)} = A^{-1}\mathbf{r}^{(k)} \cdot \mathbf{r}^{(k)} - \frac{(\mathbf{r}^{(k)} \cdot \mathbf{r}^{(k)})^2}{A\mathbf{r}^{(k)} \cdot \mathbf{r}^{(k)}}$$

5. En déduire que l'on a

$$\frac{A^{-1}\mathbf{r}^{(k+1)} \cdot \mathbf{r}^{(k+1)}}{A^{-1}\mathbf{r}^{(k)} \cdot \mathbf{r}^{(k)}} \leq \frac{(\lambda_n - \lambda_1)^2}{(\lambda_n + \lambda_1)^2}.$$

6. Soit $\mathbf{x}^*$ la solution du système : $A\mathbf{x} = \mathbf{b}$. On note $\|\cdot\|_A$ la norme associée au produit scalaire $(\cdot,\cdot)_A$ de $\mathbb{R}^n$ défini par : $(\mathbf{x},\mathbf{y})_A = A\mathbf{x}\cdot\mathbf{y}$, pour tout vecteur $\mathbf{x}$ et $\mathbf{y}$ de $\mathbb{R}^n$. Montrer que

$$\|\mathbf{x}^* - \mathbf{x}^{(k)}\|_A \leq \left(\frac{\kappa_2(A)-1}{\kappa_2(A)+1}\right)^k \|\mathbf{x}^* - \mathbf{x}^{(0)}\|_A,$$

où $\kappa_2(A)$ est le conditionnement de la matrice $A$ relatif à la norme euclidienne sur $\mathbb{R}^n$.

On utilisera notamment pour cet exercice le résultat suivant : soit $A \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive. On note $\lambda_i$, $1 \leq i \leq n$, les valeurs propres de $A$ rangées par ordre croissant :

$$0 < \lambda_1 \leq ... \leq \lambda_n.$$

Alors, nous avons l'inégalité de Kantorovitch

$$\|\mathbf{x}\|^4 \leq (A\mathbf{x}\cdot\mathbf{x})(A^{-1}\mathbf{x}\cdot\mathbf{x}) \leq \frac{(\lambda_1+\lambda_n)^2}{4\lambda_1\lambda_n}\|\mathbf{x}\|^4.$$

**Exercice 4.3** (Méthode du gradient conjugué).

Soit $A \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive et $\mathbf{b}$ dans $\mathbb{R}^n$. On désigne par $\mathbf{x}^*$ la solution du système linéaire associé à $A$ et $\mathbf{b}$. Pour tout vecteur $\mathbf{x}$, on introduit la norme

$$\|\mathbf{x}\|_A = \sqrt{A\mathbf{x}\cdot\mathbf{x}}.$$

On pose $J$ comme la fonctionnelle quadratique définie par

$$J(\mathbf{x}) = \frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}.$$

1. Soit $K$ un convexe fermé non vide de $\mathbb{R}^n$. Justifier que les problèmes de minimisation

$$\inf\{J(\mathbf{x}); \mathbf{x} \in K\}$$

et

$$\inf\{\|\mathbf{x}^* - \mathbf{x}\|_A; \mathbf{x} \in K\}$$

possédent tous deux une seule et même solution $\mathbf{x}_K \in K$. En déduire que $\mathbf{x}^*$ est la solution du problème de minimisation

$$\inf\{J(\mathbf{x}); \mathbf{x} \in \mathbb{R}^n\}.$$

2. Etant donné un point initial $\mathbf{x}^{(0)}$ tel que $\mathbf{x}^{(0)} \neq \mathbf{x}^*$. On pose $\mathbf{r}^{(0)}$ comme le résidu initial et, pour tout $k \in \mathbb{N}^*$, on désigne par $K^{(k)}$ le sous-espace vectoriel défini par

$$K^{(k)} := \text{Vect}\left\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, ..., A^k\mathbf{r}^{(0)}\right\}.$$

Cet espace est appelé sous-espace de Krylov d'ordre $k$ engendré par $\mathbf{r}^{(0)}$.

(a) Montrer que

$$\|\mathbf{x}^* - \mathbf{x}^{(k)}\|_A = \inf\left\{\|P(A)(\mathbf{x}^* - \mathbf{x}^{(0)})\|_A; P \in \mathbb{R}_k[X], P(0) = 1\right\},$$

où $\mathbb{R}_k[X]$ désigne l'espace vectoriel des polynômes à coefficients dans $\mathbb{R}$ de degré inférieur ou égal à $k$.

(b) On note $\sigma(A)$ le spectre de $A$. Montrer que, pour tout $P \in \mathbb{R}_k[X]$ tel que $P(0) = 1$, on a

$$\|\mathbf{x}^* - \mathbf{x}^{(k)}\|_A \leq \max_{\lambda \in \sigma(A)} |P(\lambda)| \|\mathbf{x}^* - \mathbf{x}^{(0)}\|_A.$$

(c) En déduire que $\mathbf{x}^{(n)} = \mathbf{x}^*$.

3. On note $0 < \lambda_1 \leq ... \leq \lambda_n$ les valeurs propres de $A$.

(a) Vérifier que, pour tout vecteur $\mathbf{z}$ de $\mathbb{R}^n$, on a l'encadrement

$$\sqrt{\lambda_1}\|\mathbf{z}\|_A \leq \|A\mathbf{z}\| \leq \sqrt{\lambda_n}\|\mathbf{z}\|_A.$$

En déduire que, pour tout indice $k$ tel que $0 < k \leq n$, on a les inégalités

$$\|\mathbf{b} - A\mathbf{x}^{(k)}\| \leq \sqrt{\kappa_2(A)}\|\mathbf{r}^{(0)}\| \frac{\|\mathbf{x}^* - \mathbf{x}^{(k)}\|_A}{\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_A},$$

où $\kappa_2(A)$ est le conditionnement de la matrice $A$ relatif à la norme euclidienne $\|\cdot\|$.

(b) Soit $T_k$ le polynôme de Tchebytchev de degré $k$. On pose

$$Q_k(x) := \frac{T_k(\frac{\lambda_n + \lambda_1 - 2x}{\lambda_n - \lambda_1})}{T_k(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1})}.$$

Montrer que, pour tout $\lambda \in \sigma(A)$, on a

$$|Q_k(\lambda)| \leq 2\left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right)^k.$$

En déduire que, pour tout entier $k$ satisfaisant $0 < k \leq n$, nous avons

$$\|\mathbf{x}^* - \mathbf{x}^{(k)}\|_A \leq 2\left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right)^k \|\mathbf{x}^* - \mathbf{x}^{(0)}\|_A.$$

**Rappel.** On rappelle que

$$T_k(x) := \cos(k \arccos x),$$

pour $|x| \leq 1$. De plus, on peut montrer que

$$T_k(x) = \frac{1}{2}(x + \sqrt{x^2-1})^k + \frac{1}{2}(x - \sqrt{x^2-1})^k,$$

pour $x \geq 1$.

**Exercice 4.4** (Méthode de Newton).

1. Vérifier que la méthode de Newton appliquée au calcul de l'inverse d'un scalaire $\alpha$ donne la méthode itérative suivante : $x^{(0)}$ est donné et on calcule : $x^{(k+1)} = x^{(k)}(2 - \alpha x^{(k)})$, pour $k \geq 0$.
2. Donner, par analogie, un algorithme qui permet de calculer l'inverse d'une matrice $A$ donnée.

**Exercice 4.5**

Soient $\mathbf{a}$, un vecteur donné de $\mathbb{R}^n$, $B$ et $C$, deux matrices carrées réelles de taille $n$. On suppose de plus $B$ inversible. Pour tout élément $\mathbf{v} \in \mathbb{R}^n$, on pose :

$$\begin{aligned} q(\mathbf{v}) &= <\mathbf{a}, \mathbf{v}> + \frac{1}{2}\|B\mathbf{v}\|^2; \\ g(\mathbf{v}) &= \frac{1}{3}\|C\mathbf{v}\|^3; \\ J(\mathbf{v}) &= g(\mathbf{v}) + q(\mathbf{v}) = <\mathbf{a}, \mathbf{v}> + \frac{1}{2}\|B\mathbf{v}\|^2 + \frac{1}{3}\|C\mathbf{v}\|^3. \end{aligned}$$

1. Démontrer que $q$ est une fonction de classe $\mathcal{C}^\infty$ sur $\mathbb{R}^n$, puis calculer la différentielle de $q$ prise en $\mathbf{v}$, dans la direction $\mathbf{h} \in \mathbb{R}^n$, ainsi que la matrice hessienne de $q$ prise en $\mathbf{v}$.
2. (a) À l'aide d'une majoration très simple et d'un petit raisonnement, prouver que la fonctionnelle $g$ est deux fois différentiable en $0_{\mathbb{R}^n}$ et que les deux premières dérivées sont nulles.
   (b) Déterminer la quantité $< g'(\mathbf{v}), \mathbf{h} >$, la différentielle de $g$ prise au point $\mathbf{v}$, dans la direction $\mathbf{h}$. *Pour cela, on pourra au choix utiliser un développement de Taylor ou calculer la différentielle de g au sens de Gâteaux.*
   (c) Démontrer que la dérivée seconde de $g$ prise en $\mathbf{v}$, dans la direction $\mathbf{h}$ a pour expression :

$$< g''(\mathbf{v})\mathbf{h}, \mathbf{h} >= \|C\mathbf{v}\|.\|C\mathbf{h}\|^2 + \frac{1}{\|C\mathbf{v}\|} < C\mathbf{v}, C\mathbf{h} >^2 .$$

3. Déduire des questions précédentes que la fonctionnelle $J$ est $\alpha$-elliptique, c'est-à-dire que : $\forall \mathbf{v} \in \mathbb{R}^n$ et $\forall \mathbf{h} \in \overrightarrow{\mathbb{R}}^n$, $< J''(\mathbf{v})\mathbf{h}, \mathbf{h} >\geq \alpha\|\mathbf{h}\|^2$. On donnera explicitement la constante $\alpha$.
   En déduire que $J$ est convexe.
4. Démontrer que le problème :

$$(\mathcal{P}) \begin{cases} \min J(\mathbf{v}) \\ \mathbf{v} \in \mathbb{R}^n \end{cases}$$

possède une solution unique.

## 4.9 Travaux pratiques

### 4.9.1 Travaux pratiques 1

*L'objectif de cette séance de travaux pratiques est de vous faire programmer quelques méthodes d'Optimisation sans contrainte de fonctionnelles. Dans un premier temps, nous nous intéreserons aux fonctions de $\mathbb{R}$ dans $\mathbb{R}$, puis aux fonctions de $\mathbb{R}^n$ dans $\mathbb{R}$*

**Exercice 4.6** (Energie rayonnante d'un corps noir).

L'énergie rayonnante d'un corps noir dans l'intervalle d'émission $[\lambda, \lambda + d\lambda]$, par unité de surface et de temps, est appelée **émittance monochromatique** maximale du corps noir et est notée $M(\lambda)$. Sa valeur, exprimée en $Wb/m^2$ est donnée par la loi de Planck :

$$M(\lambda) = \frac{2\pi h C_0^2}{n^2 \lambda^5} . \frac{1}{\exp\left\{\frac{hC_0}{nkT\lambda}\right\} - 1}.$$

Les constantes intervenant dans cette loi sont :

- $C_0 \simeq 2,997.10^8\ m/s$ : vitesse de la lumière dans le vide.
- $h \simeq 6,625.10^{-34}\ J.s$ : constante de Planck.
- $k \simeq 1,380.10^{-23}\ J/K$ : constante de Boltzmann.
- $\lambda$ : longueur d'onde (m).
- $T$ : température absolue de la surface du corps noir (K).
- $n = 1$ : indice de réfraction du milieu (ici le vide).

1. Tracer sur un même graphique la fonction $M(\lambda)$ pour les valeurs suivantes de $T$ (K) : 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800. Associer chaque courbe tracée à la valeur de $T$ correspondante. On prendra $\lambda \in [10^{-7}, 2.10^{-5}]$.
   **Fonctions Matlab utiles :** *hold on, hold off.*
2. On souhaite trouver la valeur $\lambda^*$ de $\lambda$ qui maximise l'émittance monochromatique pour une température de surface $T$ donnée. à quelle contrainte est-on soumis si l'on souhaite utiliser la méthode de la section dorée ?
   Programmer alors cette méthode pour déterminer $\lambda^*$ suivant les différentes valeurs de $T$.
3. Vérifier les lois de Wien :

   $$\lambda^* T = A \text{ et } M(\lambda^*) = BT^5, \text{ où } A \text{ et } B \text{ désignent des constantes.}$$

4. Reprendre la question 2 en utilisant cette fois la fonction préprogrammée de Matlab *fminsearch*. Que pensez-vous de la sensibilité de la méthode sur le point de départ ?

**Exercice 4.7** (Méthodes de gradient pour des fonctionnelles quadratiques).

Soit $n \in \mathbb{N}^*$. On considère la matrice $A \in \mathcal{M}_n(\mathbb{R})$ et le vecteur $\mathbf{b}_n \in \mathbb{R}^n$ définis par :

$$A_n = \begin{pmatrix} 4 & -2 & 0 & \dots & 0 \\ -2 & 4 & -2 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -2 & 4 & -2 \\ 0 & \dots & 0 & -2 & 4 \end{pmatrix} \text{ et } \mathbf{b}_n = (1, 1, \dots, 1).$$

On cherche à minimiser dans $\mathbb{R}^n$, par différentes méthodes, la fonctionnelle :

$$J_n(\mathbf{x}) = \frac{1}{2}(A_n\mathbf{x}, \mathbf{x}) - (\mathbf{b}_n, \mathbf{x}).$$

On appelle donc $(\mathcal{P}_n)$ le problème :

$$(\mathcal{P}_n) \begin{cases} \min J_n(\mathbf{x}) \\ \mathbf{x} \in \mathbb{R}^n \end{cases}$$

**Remarque :** il est important d'exploiter, dans les questions qui suivent, le format "creux" de la matrice $A$ afin de diminuer les temps de calcul.
**Fonctions Matlab utiles :** *sparse, full.*

1. Programmer en Matlab la fonctionnelle $J_n$, et la représenter dans le cas $n = 2$ sur le pavé $[-10, 10] \times [-10, 10]$.
   **Fonctions Matlab utiles :** *meshgrid, mesh.*
2. Vérifier numériquement, pour certaines valeurs de $n$ que $A_n$ est définie positive.
   Calculer la solution théorique du problème $(\mathcal{P}_n)$ dans le cas $n = 2$.
   **Fonction Matlab utile :** *eig.*
3. Nous allons étudier trois méthodes de minimisation. Pour chacune de ces études, on demande :

   **⟶ pour le cas n = 2 :**
   - d'afficher sur une même figure, et dans le cas $n = 2$, les courbes de niveau de $J_n$, et son gradient.
     **Fonctions Matlab utiles :** *contour, quiver.*
   - pour un point de départ $x^0$, de stocker la liste des $x^n$ obtenu, avant que le critère de convergence soit atteint.
   - de tracer, sur la même courbe que précédemment, les lignes qui relient les $x^n$.

   **⟶ lorsque n prend les valeurs 10, 20, 30, 50, 100 :**
   - de tester chacune des trois méthodes
   - Enfin, de comparer à l'aide d'un graphique ou d'un tableau, la rapidité de convergence de chacune de

ces méthodes, ainsi que le temps de calcul par Matlab, suivant les différentes valeurs prises par $n$.

(a) **La méthode du gradient à pas fixe.**
Écrire une fonction Matlab prenant en argument $\rho > 0$, un pas fixe et $\mathbf{x}^0 \in \mathbb{R}^n$, un vecteur d'initialisation, afin de mettre en œuvre l'algorithme du gradient à pas fixe, puis la tester sur $J_n$ dans chacun des cas ci-dessus. Répondre aux questions initiales.
Expliquer brièvement pourquoi il est important de choisir le pas fixe, ni trop grand, ni trop petit.

(b) **La méthode du gradient à pas optimal.**

i. Soient $x \in \mathbb{R}^n$, un point, et $d \in \mathbb{R}^n$, un vecteur.
Écrire une fonction Matlab permettant de minimiser, à l'aide de la méthode de la section dorée, la fonction $t \longmapsto J_n(\mathbf{x} + t\mathbf{d})$.

ii. Voici l'algrithme du gradient à pas optimal pour la minimisation d'une fonction $f$ donnée :

$$\begin{cases} \mathbf{x}^0 \text{ est donné.} \\ \mathbf{x}^{n+1} = \mathbf{x}^n + \rho_n \mathbf{d}^n \\ \mathbf{d}^n = -\nabla f(\mathbf{x}^n) \\ \rho_n = \min\limits_{\alpha \in \mathbb{R}} J_n(\mathbf{x}^n + \alpha \mathbf{d}^n). \end{cases}$$

Programmer cet algorithme et répondre aux questions initiales. On pourra utiliser la méthode de la section dorée pour calculer $\rho_n$.

(c) **La méthode du gradient conjugué dans le cas d'une fonctionnelle quadratique elliptique.**
Soit

$$\begin{array}{rccl} f: & \mathbb{R}^n & \longrightarrow & \mathbb{R} \\ & \mathbf{x} & \longmapsto & \frac{1}{2}(A\mathbf{x}, \mathbf{x}) - (\mathbf{b}, \mathbf{x}) \end{array}$$

où $A \in \mathcal{M}_n(\mathbb{R})$ est une matrice symétrique définie positive et $\mathbf{b}$ est un vecteur de $\mathbb{R}^n$. Alors, dans ce cas, l'algorithme du gradient conjugué s'écrit :

$$\begin{cases} \mathbf{x}^0 \text{ est donné, } \mathbf{r}^0 = A\mathbf{x}^0 - \mathbf{b} \text{ et } \mathbf{d}^0 = -\mathbf{r}^0. \\ \mathbf{x}^{n+1} = \mathbf{x}^n + \rho_n \mathbf{d}^n, \ \rho_n = -\dfrac{(\mathbf{r}^n, \mathbf{d}^n)}{(A\mathbf{d}^n, \mathbf{d}^n)} \\ \mathbf{r}^{n+1} = A\mathbf{x}^{n+1} - \mathbf{b} \\ \mathbf{d}^{n+1} = -\mathbf{r}^{n+1} + \beta_n \mathbf{d}^n, \ \beta_n := \dfrac{\|\mathbf{r}^{n+1}\|^2}{\|\mathbf{r}^n\|^2}. \end{cases}$$

Programmer cet algorithme et répondre aux questions initiales.

### 4.9.2 Travaux pratiques 2

*L'objectif de cette séance de travaux pratiques est d'étendre les algorithmes étudiés au cours de la séance précédente à des fonctionnelles non quadratiques, et pour les fonctionnelles quadratiques, d'étudier l'influence*

*d'un mauvais conditionnement de la matrice Hessienne.*

**Quelques brefs rappels d'Analyse Numérique.**

CONDITIONNEMENT D'UNE MATRICE

On souhaite résoudre le système $A\mathbf{X} = \mathbf{b}$, avec :

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}.$$

– Si $\mathbf{b} = \begin{pmatrix} 32 \\ 33 \\ 33 \\ 31 \end{pmatrix}$.

Il est clair que la solution de ce système est $\mathbf{X} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$.

– Si on modifie **légèrement b** de la façon suivante : $\mathbf{b} = \begin{pmatrix} 32,1 \\ 32,9 \\ 33,1 \\ 30,9 \end{pmatrix}$.

Dans ce cas, la solution du système est grandement modifiée, puisque l'on trouve par des techniques usuelles : $\mathbf{X} = \begin{pmatrix} 92 \\ -12,6 \\ 4,5 \\ -11 \end{pmatrix}$.

Ainsi, une toute petite modification de **b** engendre une grande modification de **X**. On se rend ainsi compte qu'une imprécision dans un calcul numérique peut conduire à des résultats erronés. On dit que la matrice $A$ est **mal conditionnée**. De façon plus précise, on rappelle la définition :

**Définition 18** *Conditionnement d'une matrice.*
*Soit $A \in \mathcal{M}_{n,n}(\mathbb{R})$. On note $\|.\|$, la norme subordonnée à la norme euclidienne. On définit $\kappa(A)$, le conditionnement de la matrice $A$ par :*

$$\kappa(A) = \|A\|.\|A^{-1}\|.$$

**Exercice 4.8** (Minimisation d'une fonctionnelle non quadratique).

On souhaite résoudre, dans cet exercice, un problème de minimisation d'une fonctionnelle non linéaire et non quadratique. On définit sur $\mathbb{R}^2$ la fonctionnelle de *Rosenbrock*, également appelée *Rosenbrock banana*, par la relation :

$$f(x, y) = (x - 1)^2 + 10(x^2 - y)^2.$$

1. **Étude théorique.**
   (a) Trouver les points critiques de $f$.
   (b) Démontrer que $f$ admet un unique minimum global qu'elle atteint en $\mathbf{X}^\star := (1, 1)$.
   (c) On appelle $H$, la matrice hessienne de $f$ calculée en $\mathbf{X}^\star$.
   Déterminer $H$, puis calculer son conditionnement.
   **Fonction Matlab utile :** *cond*.
   (d) Écrire un développement limité à l'ordre 2 de la fonction $f$ en $\mathbf{X}^\star$.
   Expliquer pourquoi ce mauvais conditionnement risque de gêner la convergence des algorithmes numériques. Une explication intuitive sera acceptée.
2. **Étude Numérique.**
   (a) Représenter la surface représentative de la fonctionnelle de *Rosenbrock* sur le pavé $[-10, 10] \times [-10, 10]$.
   (b) Tracer, dans le plan $xOy$, cent lignes de niveaux de la fonctionnelle de *Rosenbrock* dans le pavé $[0, 2] \times [0, 2]$.
   Quel inconvénient numérique risque t-on de rencontrer ?
3. On souhaite comparer deux méthodes bien connues de minimisation. La méthode de gradient à pas fixe, puis à pas optimal. Pour comparer les deux méthodes, on choisit de démarrer les algorithmes que l'on codera au point de coordonnées $(0, 1)$.
   (a) **Méthode du gradient à pas optimal.**
      i. Écrire une fonction *pasOptimal.m*, des deux variables $(\alpha, \mathbf{X}) \in \mathbb{R} \times \mathbb{R}^2$ qui renvoie la quantité $f(\mathbf{X} - \alpha \nabla f(\mathbf{X}))$. On minimisera par la suite la première variable de cette fonction, à l'aide de la fonction Matlab *fminsearch*.
      ii. Écrire un programme mettant en œuvre la méthode du gradient à pas optimal pour minimiser la fonctionnelle de Rosenbrock. On utilisera un double critère d'arrêt, après l'avoir justifié : un critère portant sur le nombre total d'itérations de la méthode, et un autre critère relatif à la précision de la méthode.
      **Fonction Matlab utile :** *norm*.
      iii. Compléter alors le programme précédent afin de représenter cent lignes de niveaux de la fonctionnelle de *Rosenbrock*, et la courbe reliant les points obtenus à chaque itération par la méthode précédente.
   (b) **Méthode du gradient à pas fixe.**
   Écrire un programme *PasFixe.m* mettant en œuvre la méthode du gradient à pas fixe. Ce programme prendra comme arguments $\mathbf{X}$ et $\beta$, où $\mathbf{X}$ désigne le point de démarrage de la méthode et $\beta$, le pas de la méthode. Pour tester cette méthode, on pourra choisir par exemple $\beta = 0,01$.

(c) **Comparaison des méthodes.**

i. Sur la même figure que précédemment, représenter les itérés obtenus par la méthode du gradient à pas fixe.

ii. Quelle remarque pouvez-vous faire sur les lignes de niveaux de la fonctionnelle $f$, dans un voisinage du point critique $\mathbf{X}^\star$. Proposez une explication du phénomène observé pour la méthode du gradient à pas optimal.

iii. Pour l'une ou l'autre des méthodes, on appelle $\mathbf{X}_k$, les points obtenus en appliquant les algorithmes à chaque itération.
Tracer sur un graphe différent, et pour chacune des deux méthodes, la courbe donnant $\ln(\|\mathbf{X}_k - \mathbf{X}^\star\|)$ en fonction de $k$. Pour quel type de fonction cette courbe est-elle une droite ?

**Exercice 4.9** (Résolution de systèmes linéaires à l'aide de préconditionneurs).

On souhaite comparer, dans cette exercice, deux méthodes numériques de résolution de systèmes linéaires de grande taille, utilisables dans le cas de matrices creuses. On va réutiliser la famille de matrices introduite dans le TP n1. Pour $n \in \mathbb{N}^*$, on appelle $A_n$, la matrice carrée de taille $n \times n$ définie par :

$$A_n = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}.$$

On souhaite résoudre le système d'inconnue $\mathbf{X} \in \mathcal{M}_{n,1}(\mathbb{R})$ : $A_n\mathbf{X} = \mathbf{b}$, où $\mathbf{b}_n$ est un vecteur colonne don de taille $n$ donné. Pour les tests numériques, on pourra par exemple choisir :

$$\mathbf{b}_n = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

1. **Première méthode : le gradient conjugué.**
On rappelle l'algorithme du gradient conjugué pour une fonctionnelle quadratique :

$$\begin{cases} \mathbf{x}^0 \text{ est donné, } \mathbf{r}^0 = A\mathbf{x}^0 - \mathbf{b} \text{ et } \mathbf{d}^0 = -\mathbf{r}^0 \ ; \\ \mathbf{x}^{n+1} = \mathbf{x}^n + \rho_n \mathbf{d}^n, \ \rho_n = -\dfrac{(\mathbf{r}^n, \mathbf{d}^n)}{(A\mathbf{d}^n, \mathbf{d}^n)} \ ; \\ \mathbf{r}^{n+1} = A\mathbf{x}^{n+1} - \mathbf{b} \ ; \\ \mathbf{d}^{n+1} = -\mathbf{r}^{n+1} + \beta_n \mathbf{d}^n, \ \beta_n = \dfrac{\|\mathbf{r}^{n+1}\|^2}{\|\mathbf{r}^n\|^2}. \end{cases}$$

En utilisant une fonctionnelle quadratique **judicieusement choisie**, programmer l'algorithme de gradient conjugué pour déterminer la solution du système $A_n\mathbf{X} = \mathbf{b}_n$.

⟶ L'entier $n$ sera placé dans les arguments d'entrée du programme, de même que $\mathbf{x}^0$, qui permet d'initialiser l'algorithme et $\mathbf{b}_n$, le second membre du système à résoudre.

⟶ On testera le programme pour différentes valeurs de $n$, en allant au maximum jusqu'à $n = 1000$.

⟶ On utilisera un double critère pour stopper l'algorithme : un critère sur le nombre maximal d'itération et un autre critère sur la norme du résidu. On notera à chaque fois le nombre d'itérations nécessaires pour atteindre la solution.

⟶ On pourra comparer la solution trouvée par ce programme avec la solution réelle de ce système calculée par Matlab.

2. **La méthode du gradient conjugué préconditionné.**
   Cette méthode sert à réduire le nombre d'itérations de l'algorithme. L'idée de cette méthode est basée sur la remarque suivante : si $M$ est une matrice inversible, alors, la solution du système $A_n\mathbf{X} = \mathbf{b}_n$ est la solution du système $M^{-1}A_n\mathbf{X} = M^{-1}\mathbf{b}_n$. Des difficultés numériques peuvent survenir si le conditionnement de la matrice $A$ est mauvais. On va donc choisir $M$ pour que le conditionnement de $M^{-1}A_n$ soit meilleur que le conditionnement de $A_n$, et pour que l'inverse de $M$ soit aisée à calculer.

   (a) Expliquer brièvement le principe de factorisation de Cholesky.

   (b) $A_n$ étant une matrice creuse, on note $RI$, la factorisée incomplète de Cholesky de $A_n$, et on pose $M = {}^tRI \times RI$. Vérifier que le conditionnement de la matrice $M^{-1}A_n$ est meilleur que celui de $A_n$ et que l'inverse de $M$ se calcule aisément.
   **Fonction Matlab utile :** *inv*, *cholinc*(avec l'option $'0'$).

   (c) En utilisant les remarques précédentes, améliorer (on réécrira un programme) le programme précédent pour calculer la solution du système $A_n\mathbf{X} = \mathbf{b}$. On tiendra compte des remarques faites pour la méthode du gradient conjugué sans préconditionnement.

3. **Comparaison des deux méthodes.**
   Compléter les deux programmes précédents afin de tracer, à chaquel appel des programmes le résidu logarithmique en fonction du nombre d'itérations. Conclure.

# Chapitre 5

# Quelques algorithmes pour l'optimisation avec contraintes

Nous avons traité, dans le chapitre précédent, de la résolution de problèmes de minimisation sans contraintes.
Nous considérons maintenant le cas avec contraintes.
Plus précisément, soit $C$ un ensemble non vide, fermé de $\mathbb{R}^n$ et on s'intéresse à la résolution du problème

$$(\mathcal{P}) \qquad \begin{array}{c} \min J(x) \\ x \in C \end{array}$$

## 5.1 Retour sur les conditions d'optimalité

Rappelons le théorème de Kuhn-Tucker :
Soit $J$, $f_i, i \in \{1, ..., p\}$, $g_i, i \in \{1, ..., m\}$ des fonctions de classe $\mathcal{C}^1$. On veut minimiser $J$ sur l'ensemble :

$$C = \left\{ X \in \mathbb{R}^N, f_i(X) = 0, i \in \{1, ..., p\}, g_i(X) \leqslant 0, i \in \{1, ..., m\} \right\}.$$

On suppose les contraintes qualifiées au point $X^*$. Alors, une condition est nécessaire pour que $X^*$ soit un minimum de $J$ est qu'il existe des nombres positifs $\lambda_1^*, ..., \lambda_m^*$, et des nombres réels $\mu_1^*, ..., \mu_p^*$ tels que

$$(1) \quad \left\{ \begin{array}{c} \nabla J(X^*) + \sum^{m} \lambda_i^* \nabla g_i(X^*) + \sum_{i=1}^{p} \mu_i^* \nabla f_i(X^*) = 0 \\ \text{avec } \lambda_i^* g_i(X^*)^{i=1} = 0, \quad 1 \leq i \leq m. \end{array} \right.$$

Introduisons maintenant la définition suivante :

**Définition 19** *On appelle Lagrangien du problème* $(\mathcal{P})$ *la fonction définie sur* $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m$ *par :*

$$\mathcal{L}(x, \mu, \lambda) = J(x) + \sum_{i=1}^{p} \mu_i \ f_i(x) + \sum_{j=1}^{m} \lambda_i g_i(x)$$

*La relation* (1) *s'écrit alors :*

$$\nabla_x \mathcal{L}(x^*, \mu^*, \lambda^*) = 0$$

On a alors le théorème important suivant dans le cas convexe.

**Theorem 5.1** *(CNS dans le cas convexe).*

*On suppose que* $J$, $f$ *et* $g$ *sont* $\mathcal{C}^1$, *que* $J$, $g$ *sont convexes,* $f$ *est affine et que* $x^*$ *est régulier pour les contraintes* $f$ *et* $g$, *c'est-à-dire :*

- *qu'il est réalisable* $f_i(x^*) = 0$, $g_i(x^*) \leqslant 0$.
- *et que les vecteurs* $\nabla f_i(x^*)$, $\nabla g_j(x^*)$, *avec* $1 \leqslant i \leq p$, $j \in I_0(x^*)$, *sont indépendants.*

*Alors* $x^*$ *est une solution du problème* $(\mathcal{P})$ *si, et seulement si les conditions du théorème précédent de Kuhn-Tucker sont satisfaites.*

## 5.2 Conditions d'optimalité nécessaires du second ordre

Les résultats énoncés jusqu'à présent donnent des conditions nécessaires pour résoudre le problème de minimisation sous contraintes. Ils fournissent en soi des candidats valables pour résoudre $(\mathcal{P})$, c'est-à-dire les points critiques du Lagrangien.

Toutefois, il est nécessaire, pour pouvoir conclure, d'avoir des résultats précisant si la solution obtenue est effectivement un minimum. Pour cela, on peut, grâce à une conditon du second ordre, restreindre le nombre de candidats. C'est l'objet du théorème suivant.

**Theorem 5.2** *On suppose que* $J, f$ *et* $g$ *sont de classe* $\mathcal{C}^2$, *que* $x^*$ *est un minimum (local) de* $J$ *sur* $C$ *et que le point* $x^*$ *est régulier. Alors,*

- $\mu^* = \left(\mu_1^*, ..., \mu_p^*\right), \lambda^* = (\lambda_1^*, ..., \lambda_m^*)$ *telles que*
- *les relations de Kuhn-Tucker soient satisfaites*
- *pour toute direction* $d \in \mathbb{R}^N$ *vérifiant :*

$$(\nabla f_i(x^*), d) = 0, i = 1, ..., p$$

$$(\nabla g_j(x^*), d) = 0, j \in I_0^+(x^*)$$

$$(\nabla g_j(x^*), d) \leqslant 0, j \in I_0(x^*) \setminus I_0^+(x^*)$$

*où*

$$I_0^+(x^*) = \left\{j, 1 \leqslant j \leqslant q / gj(x^*) = 0 \text{ et } \mu_j^* > 0\right\}$$

```
Initialisation
k = 0 ; choix de x_0 et g_0 > 0
Iteration k
Tant que le critere d'arret est non satisfait
      x̂^(k+1) = x^(k) − f^(k) ∇J(x^(k))
      x^(k+1) = Π_C x̂^(k+1)
      k = k + 1
fin
```

TAB. 5.1 – Algorithme du gradient projeté.

*on a :*

$$\left(\nabla^2_{xx}\mathscr{L}\left(x^*,\mu^*,\lambda^*\right)d,d\right)\geqslant 0$$

$\nabla^2_{xx}\mathscr{L}\left(x,\mu,\lambda\right)$ *désigne la dérivée seconde de* $\mathscr{L}$ *au point* $(x,\mu,\lambda)$.

**Définition 20** *L'ensemble* $I_0^+\left(x^*\right)$ *est l'ensemble des contraintes fortement actives. Lorsque* $I_0^+\left(x^*\right)=I_0\left(x^*\right)$, *c'est-à-dire* $0=g_j\left(x^*\right)\Longleftrightarrow\lambda_j^*>0$, *on dit qu'il y a stricte complémentarité.*

## 5.3 Méthode du gradient projeté

Rappelons que, dans le cas sans contrainte, l'algorithme du gradient, qui est une méthode de descente, s'écrit sous la forme générique.

$$\left\{\begin{array}{l} x^{(0)}\in\mathbb{R}^n \ \ \textit{donné.}\\ x^{(k+1)}=x^{(k)}+f^{(k)}d^{(k)}, d^{(k)}\in\mathbb{R}^n\setminus\{0\}, f^{(k)}\in\mathbb{R}^{+*}\end{array}\right.$$

où $f^{(k)}$ et $d^{(k)}$ sont choisis de telle sorte que $J\left(x^{(k+1)}\right)\leqslant J\left(x^{(k)}\right)$. Lorsque l'on minimise sur un ensemble de contraintes $C$, il n'est pas sûr que $x^{(k)}$ reste sur $C$. Il est donc nécessaire de se ramener sur $C$. On réalise cette dernière opération grâce à une projection sur $C$, l'opérateur associé étant noté $\Pi_e$ : $\begin{array}{ccc} \mathbb{R}^n & \rightarrow & C \\ x & \longmapsto & \Pi_C\left(x\right)\end{array}$.

Ceci donne lieu alors naturellement à l'algorithme du gradient projeté, algorithme identique à celui du gradient à la projection près.

Nous avons alors le résultat de convergence suivant :

**Theorem 5.3** *Soit* $J$ *une fonction* $\mathcal{C}^1$ *de* $\mathbb{R}^n$ *dans* $\mathbb{R}$. *On suppose que* $J$ *est elliptique de dérivée lipschitzienne. Alors, si on choisit le pas* $f^{(k)}$ *dans un intervalle* $[\beta_1;\beta_2]$ *tel que* $0<\beta_1<\beta_2<\frac{2\alpha}{M}$ *(où* $\alpha$ *et* $M$ *sont respectivement les constantes d'elliplicité (*$(\nabla J(x)-\nabla J(y),x-y)\geq\alpha\|x-y\|^2$*) et d'inégalité lipschitzienne*

($\|J(x) - J(y)\| \leq M\|x - y\|$). *La suite* $\left(x^{(k)}\right)_k$ *définie par la méthode du gradient projeté converge vers la solution du problème* $(\mathcal{P})$.

Cette approche paraît simple au premier abord. Toutefois, il ne faut pas oublier que l'on doit connaître l'opérateur de projection sur $C$, ce qui n'est pas, à priori simple. Il est clairement hors de question de résoudre le problème de minimisation pour $X \in \mathbb{R}^n$ fixé :

$$\begin{cases} \min \|Y - X\|^2 \\ Y \in C \end{cases}$$

qui est lui-même un problème de minimisation sur le même ensemble de contraintes. Dans quelques cas particuliers, on peut expliciter l'opérateur de projection.

Supposons que $C$ soit une intersection de demi espaces du type :

$$C = \{x = (x_1, ..., x_n), x_i \geqslant a_i, i \in I, x_j \leqslant b_j, j \in J\}$$

$I$ et $J$ étant des ensembles d'indices non nécessairement disjoints (ceci contient notamment le cas des pavés).

Pour ce qui est du cas d'une contrainte du type $x_i \geqslant a_i$, on peut se convaincre facilement que la i-ième coordonnée de $\Pi_c x$ sera $x_i$ si $x_i \geqslant a_i$ et $a_i$ sinon. On raisonne de même pour les $x_j$. On peut résumer cela par :

$$(\Pi_c x)_i = \max(x_i, a_i) = x^+ \text{ ou } (\Pi_x)_j = \min(x_j, b_j) = x^-.$$

Si on a les deux contraintes à la fois, on pose alors

$$\Pi_c x = \begin{cases} x & \text{, si } x \in C \text{ ;} \\ x_0 + R\frac{x - x_0}{\|x - x_0\|} & \text{, si } x \notin C \end{cases}$$

## 5.4 Méthode de Lagrange-Newton pour des contraintes en égalité

Plaçons-nous dans la cas particulier d'un problème quadratique avec contraintes en égalité affines.

$$\begin{cases} \min\left(\frac{1}{2}x^t Q x - c^t x\right) \\ Ax = b \end{cases}$$

où $Q$ est une matrice carrée de $\mathcal{M}_{n\times n}(\mathbb{R})$ et $c$ un vecteur de $\mathbb{R}^n$, $A \in \mathcal{M}_{p\times n}(\mathbb{R})$, $b \in \mathbb{R}^p$. Si nous écrivons les relations de Kuhn-Tucker (conditions d'optimalité au 1er ordre), nous avons :

$$\begin{cases} \nabla_x\left(\frac{1}{2}x^t Q x - c^t x + \mu^t(Ax - b)\right) = 0 \\ Ax = b \end{cases}$$

où $\mu \in \mathbb{R}^p$ est le multiplicateur de Lagrange associé à la contrainte $Ax - b = 0$. Par conséquent, le couple optimal $(x^*, \mu^*)$ est la solution du système

$$\begin{cases} Qx^* - c + A^t\mu^* = 0 \\ Ax^* = b \end{cases}$$

que l'on peut aussi écrire :

$$\begin{bmatrix} Q & A^t \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \mu^* \end{bmatrix} = \begin{bmatrix} c \\ b \end{bmatrix}$$

Si la matrice $\mathcal{M} = \begin{bmatrix} Q & A^t \\ A & 0 \end{bmatrix}$ est inversible, ce système admet une solution que l'on peut calculer par n'importe quelle méhtode pour la résolution des systèmes linéaires.

Supposons maintenant que le problème ne soit plus quadratique. On va résoudre le système d'optimalité par la méthode de Newton. Considérons le problème en égalité suivant :

$$\begin{cases} \min\ J(x) \\ h(x) = 0 \end{cases}$$

où $J : \mathbb{R}^n \to \mathbb{R}$ et $h : \mathbb{R}^n \to \mathbb{R}^p$ sont suffisamment régulières.

Les conditons du premier ordre s'écrivent (au moins formellement)

$$\begin{cases} \nabla_x \mathscr{L} L(x, \mu) = 0 \\ h(x) = 0 \end{cases}$$

où $\mu \in \mathbb{R}^p$ et $\mathscr{L}(x, \mu) = J(x) + \mu\ h(x)$ est le Lagrangien du problème. On peut alors rsoudre ce système d'équations non linéaires par la méthode de Newton.

Nous obtenons alors ce quel'on appelle la méthode de Lagrange-Newton.

Remarquons que si l'on ajoute $\nabla h\left(x^{(k)}\right)^t \mu^{(k)}$ à la première ligne de (2), on a alors la forme équivalente :

$$\begin{bmatrix} D^2_{x^x}\mathcal{L}\left(x^{(k)}, \mu^{(k)}\right) & \nabla h\left(x^{(k)}\right)^t \\ \nabla h\left(x^{(k)}\right) & 0 \end{bmatrix} \begin{bmatrix} d^{(k)} \\ \mu^{(k+1)} \end{bmatrix} = - \begin{bmatrix} \nabla J\left(x^{(k)}\right)^t \\ h\left(x^{(k)}\right) \end{bmatrix}$$

## 5.5 Méthode de Newton projetée (pour des contraintes de borne)

La méthode de Newton projetée relève d'une idée analogue à celle développée lors du grandient projeté : puisque les itérés successifs ne satisfont pas les contraintes, on les projette sur l'ensemble des contraintes. Nous

Initialisation
$k = 1$, choix de $\left(x^{(0)}, \mu^{(0)}\right) \in \mathbb{R}^n \times \mathbb{R}^p$
Iteration $k$
Tant que le critre d'arret est non satisfait
Resoudre le systeme lineaire

$$\left[ \begin{array}{cc} D^2_{xx}\mathscr{L}\left(x^{(k)}, \mu^{(k)}\right) & \nabla_x h\left(x^{(k)}\right)^t \\ \nabla_x h\left(x^{(k)}\right) & 0 \end{array} \right] \left[ \begin{array}{c} d^{(k)} \\ y^{(k)} \end{array} \right] = - \left[ \begin{array}{c} \nabla_x \mathscr{L}\left(x^{(k)}, \mu^{(k)}\right)^t \\ h\left(x^{(k)}\right) \end{array} \right] \tag{2}$$

ou $D^2_{xx}\mathcal{L}\left(x^{(k)}, \mu^{(k)}\right)$ est le Hessien par rapport a $x$ de $\mathscr{L}$.
Poser :
$x^{(k+1)} = x^{(k)} + d^{(k)}$
$\mu^{(k+1)} = \mu^{(k)} + y^{(k)}$
$k = k + 1$
fin

TAB. 5.2 – Méthode de Lagrange-Newton.

ne nous intéresserons ici qu'au cas où nous avons des contraintes de bornes de la forme :

$$(\mathcal{P}) \left\{ \begin{array}{l} \min\ f(x) \\ a \leqslant x \leqslant b. \end{array} \right.$$

Remarquons que beaucoup de problèmes duaux (où $\mu$ est alors l'inconnue) sont également de cette forme.

Commençons par un problème plus simple :

$$(\mathcal{P})_0 \left\{ \begin{array}{l} \min\ f(x) \\ x \geqslant 0 \end{array} \right.$$

où $f$ est $\mathcal{C}^2$ sur $\mathbb{R}^n$. Si $H^{(k)}$ désigne la matrice Hessienne $\left[D^2 f(x)\right]$, une itration de la méthode de Newton est de la forme :

$$x^{(k+1)} = x^{(k)} - \left(H^{(k)}\right)^{-1} \nabla f\left(x^{(k)}\right).$$

On peut affiner un peu en introduisant un pas $\alpha^{(k)} > 0$ qui donne la nouvelle formulation :

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \left[H^{(k)}\right]^{-1} \nabla f\left(x^{(k)}\right).$$

Si nous projetons sur l'ensemble des contraintes, nous obtenons alors :

$$x^{(k+1)} = \left(x^{(k)} - \alpha^{(k)} \left[H^{(k)}\right]^{-1} \left(x^{(k)}\right)\right)^+$$

où

$$(f(x))^+ = \max(f(x), 0).$$

Initialisation
Poser le n°
Choisir $x^{(0)} \in \mathbb{R}^n, x^{(0)} \geqslant 0$
Choisir une tolerance $\varepsilon > 0$
Iteration $k$ : tant que le critere d'arret est non satisfait
Application de la regle anti zig-zag :

$$w^{(k)} = \left\| x^{(k)} - \left( x^{(k)} - \nabla f\left(x^{(k)}\right)\right)^+ \right\|$$

$$\varepsilon^{(k)} = Min\left(\varepsilon, \omega^{(k)}\right)$$

$$I^{+(k)} = \left\{ i/0 \leqslant x_i^{(k)} \leqslant \varepsilon^{(k)} \ \ et \ \ \frac{\partial f}{\partial x_i}\left(x^{(k)}\right) > 0 \right\}$$

$$D^{(k)} = \left[H^{(k)}\right]^{-1} \ \ où \ \ H_{ij}^{(k)} = \begin{cases} 0 & \text{si} \quad i \neq j \quad et \quad \left(i \in I^{+(k)} \quad où \quad j \in I^{+(k)}\right), \\ \frac{\partial^2 f}{\partial x_i \partial x_j}\left(x^{(k)}\right) & \text{sinon.} \end{cases}$$

$p^{(k)} = D^{(k)} \nabla f\left(x^{(k)}\right)$ (choix de la direction de descente)

choix du pas par une recherche lineaire$x^{(k)}(\alpha) = \left(x^{(k)} - \alpha \ p^{(k)}\right)^+$

$\alpha^{(k)}$ est choisi avec une regle de type Armijo comme en (3)

$x^{(k+1)} = x^{(k)}\left(\alpha^{(k)}\right)$

fin pour i

TAB. 5.3 – Algorithme de Newton projeté : cas unilatéral

Ceci donne alors la méthode de Newton projetée.

Nous allons donner maintenant une méthode de type Quasi-Newton dont les propriétés de convergence sont proches de celles de la méthode de Newton projetée tout en étant à la fois plus souple. Nous supposons, pour que la suite des itérés soit bien définie, que les matrices $H^{(k)}$ sont inversibles. Donnons, pour commencer, la règle du choix du pas :

- Choisir $\beta \in ]0;1[\,, \sigma \in ]0;1/2[$
- Poser $\alpha^{(k)} = \beta^{m(k)}$ où $m^{(k)}$ est le plus petit entier tel que

$$f\left(x^{(k)}\right) - f\left(\beta^m \ x^{(k)}\right) \geqslant \sigma \left( \beta^m \sum_{i \notin I^{(k)+}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} p_i^{(k)} + \sum_{i \in I^{(k)+}} \frac{\partial f\left(x^{(k)}\right)}{\partial x^{(i)}} \left( x_i^{(k)} - \left(\beta^m x^{(k)}\right)_i \right) \right)$$

L'algorithme de Newton projeté dans le cas unilatéral est alors donné par la table suivante :

**Remarques :**

1. L'ensemble $I^{(k)+}$ est l'ensemble des indices des contraintes "presque" actives à $\varepsilon^{(k)}$ près. La règle anti zig-zag permet d'éviter les oscillations de l'algorithme.
2. $\left[D^{(\omega)}\right]^{-1}$ est une approxiamtion de la matrice hessienne plus "facile" à calculer.
3. La règle de choix de pas est une règle de type Arnijo qui permet de choisir un pas "optimal" à moindre coût.

Initialisation
$k = 1$
Choix de $x^{(0)}$ $\quad x^{(0)} \geqslant 0$
Choix d'une tolerance $\varepsilon > 0$
Iteration $k$
Tant que le critere de convergence n'est pas satisfait
Regle anti zig-zag
$\omega^{(k)} = \left\| x^{(k)} - \left[ x^{(k)} - \nabla f\left(x^{(k)}\right) \right]^* \right\|$
$\varepsilon^{(k)} = \min\left(\varepsilon, \omega^{(k)}\right)$
$I^{(k)\#} = \left\{ i/a_i \leqslant x_i^{(k)} \leqslant a_i + \varepsilon^{(k)} \ \ et \ \ \frac{\partial f}{\partial x_i}\left(x^{(k)}\right) > 0 \right\} \cup \left\{ i/b_i - \varepsilon^{(k)} \leqslant x_i^{(k)} \leqslant b_i \ \ et \ \ \frac{\partial f}{\partial x_i}\left(x^{(k)}\right) < 0 \right\}$
$D^{(k)}$ est definie positive et diagonale par rapport a $I^{(k)\#}$
$x^{(k)}(\alpha) = \left[ x^{(k)} - \alpha D^{(k)} \nabla f\left(x^{(k)}\right) \right]^{\#}$
$\alpha^{(k)}$ est choisi par une regle de type Armijo avec $\#$ a la place de $+$.
$x^{(k+1)} = x^{(k)}\left(\alpha^{(k)}\right)$

TAB. 5.4 – Algorithme de Newton projeté : cas bilatéral

On a alors le résultat de convergence suivant.

**Theorem 5.4** *On suppose que la fonction $f$ est convexe et $\mathcal{C}^2$ et que le problème $(P)_0$ a une unique solution $x^*$ vérifiant $\frac{\partial f}{\partial x^i}(x^*) > 0, \forall i \in I(x^*)$ (ensemble actif). On suppose, de plus, qu'on peut trouver $m_1$ et $m_2$ deux réels strictement positifs tels que :*

$$m_1 \|z\|^2 \leqslant \left(D^2 f(x)\, z, z\right) \leqslant m_2 \|z\|^2$$

*dans chacun des cas suivants :*
- *pour tout $z \in \left\{x/f(x) \leqslant f\left(x^{(0)}\right)\right\}$ d'une part ;*
- *$x$ dans une boule centrée en $x^*$ et $z \neq 0$ tel que $z_i = 0$, pour $i \in I(x^*)$ d'autre part.*

*Alors, la suite $x^{(k)}$ engendrées par l'algorithme converge vers $x^*$ et le taux de convergence est superlinéaire (au moins quadratique si $D^2 f$ est lipschitzienne au voisinage de $x^*$). On peut alors généraliser l'algorithme précédent au problème (3) qui donne l'algorithme de Newton projeté dans le cas bilatéral.*

### 5.5.1 Méthodes de pénalisation

Les méthodes de pénalisation sont très utilisées en pratique car elles sont très simples. Elles partent du principe suivant : on remplace le problème avec contraintes.

$$(\mathcal{P}) \begin{cases} \min \ J(x) \\ x \in C \subset \mathbb{R}^n \end{cases}$$

par un problème sans contraintes

$$(\mathcal{P}_\varepsilon) \begin{cases} \min\ J(x) + \frac{1}{\varepsilon}\alpha(x) \\ x \in \mathbb{R}^n \end{cases}$$

où $\alpha : \mathbb{R}^n \to \mathbb{R}$ est une fonction de pénalisation des contraintes et $\varepsilon > 0$. Le but est de trouver des fonctions $\alpha$ telles que les problèmes $(P)$ et $(P_\varepsilon)$ soient équivalents, c'est-à-dire, tels qu'ils aient les mêmes solutions. Dans ce cas, on dit que la pénalisation est exacte. On peut, par exemple, choisir :

$$\alpha(x) = \begin{cases} 0 & \text{si } x \in C \\ +\infty & \text{si } x \notin C \end{cases}$$

Cette fonction n'a pas de bonnes propriétés mathématiques (notamment la dérivabilité) pour qu'on puisse appliquer les techniques de résolution sans contraintes. Le principe intuitif des méthodes de pénalisation est que l 'on espère que, lorsque $\varepsilon$ devient petit, on aura tendance à satisfaire la contrainte $\alpha$ pour compenser.

En général, on effectue ce que l'on appelle une pénalisation dite inexacte, telle que le problème $(\mathcal{P})$ à des solutions qui ne sont pas solutions de $(\mathcal{P}_\varepsilon)$ ; l'ensemble des solutions de $(\mathcal{P}_\varepsilon)$ ne couvre pas tout l'ensemble des solutions de $(\mathcal{P})$.

Néanmoins, on peut trouver dans ce cas des fonctions $\alpha$ qui sont dérivables, ce qui permet d'utiliser les résultats de minimisation sans contraintes.

Donnons quelques exemples de fonctions de pénalisation $\alpha$ où la pénalisation est dite extérieure car la suite $(x_\varepsilon)_{\varepsilon>0}$ converge vers $x^*$ en venant de l'extérieur de $C$.

Ici, nous supposons que $\alpha$ vérifie les propriétés suivantes :

1. $\alpha$ est continue sur $\mathbb{R}^n$
2. $\forall x \in \mathbb{R}^n, \alpha(x) \geqslant 0$
3. $\alpha(x) = 0 \Leftrightarrow x \in C$

Nous donnons quelques exemples de fonction de pénalisation pour différentes contraintes :
- Contrainte $x \leqslant 0$ : la fonction $\alpha$ est $\alpha(x) = \|x^+\|^2$.
- Contrainte $h(x) = 0$ : la fonction $\alpha$ est $\alpha(x) = \|h(x)\|^2$.
- contrainte $g(x) \leqslant 0$ : la fonction $\alpha$ est $\alpha(x) = \left\|g(x)^+\right\|^2$.

où $\|.\|$ est bien sûr la norme euclidienne de $\mathbb{R}^n$ et $x^+ = \left(x_1^+, ..., x_n^+\right)$. Nous avons alors le résultat de convergence suivant :

**Theorem 5.5** *Soit $J$ une fonction continue et coercive. Soit $C$ un ensemble fermé non vide. On suppose que $\alpha$ vérifie les conditions suivantes :*

1. *$\alpha$ est continue sur $\mathbb{R}^n$.*

Initialisation
$k = 1$
Choisir $x^{(0)} \in \mathbb{R}^n, \varepsilon^{(1)} > 0$
Iteration $k$ tant que le critere d'arret n'est pas satisfait :
a) Resoudre le sous probleme $(\mathcal{P}_{\varepsilon^{(k)}})$ $\begin{cases} \min\limits_{x \in \mathbb{R}^n} J(x) + \frac{1}{\varepsilon^{(k)}} \alpha(x) \end{cases}$ avec $x^{(k-1)}$ le point d'initialisation.
b) $k \leftarrow k+1$, prendre $\varepsilon^{(k+1)} < \varepsilon^{(k)}$.

TAB. 5.5 – Algorithme de pénalisation extérieure

2. $\forall x \in \mathbb{R}^n, \alpha(x) \geqslant 0$.
3. $\alpha(x) = 0 \Leftrightarrow x \in C$.

*On a alors :*

- $\forall \varepsilon > 0, \quad (\mathcal{P}_\varepsilon)$ *a au moins une solution* $x_\varepsilon$
- *La famille* $(x_\varepsilon)_{\varepsilon>0}$ *est bornée*
- *Toute sous-suite convergente extraite de* $(x_\varepsilon)_{\varepsilon>0}$ *converge vers une solution de* $(\mathcal{P})$ *lorsque* $\varepsilon \to 0$.

On obtient alors l'algorithme suivant de pénalisation extérieure.

### 5.5.2 Méthode de dualité : méthode d'Uzawa

La technique proposée ici provient de la partie de l'optimisation appelée théorie de la dualité convexe. L'idée générale est de considérer le Lagrangien $\mathscr{L}$ au lieu de la fonction $J$ ; ce choix est motivé (au moins) par deux raisons :

- La fonction Lagrangienne englobe à la fois la fonction $J$ et les contraintes $f$ et $g$ et représente bien le problème.
- Ensuite, nous avons vu qu'une condition nécessaire du premier ordre pour que $x^*$ soit un minimum de $J$ avec contraintes est que $x^*$ (associé aux multiplicateurs de Lagrange) soit un point critique de $\mathcal{L}$.

Rappelons que le Lagrangien du problème est :

$$\mathscr{L}(x, \mu, \lambda) = J(x) + \sum_{i=1}^{p} \mu_i f_i(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

Nous avons besoin, pour la suite, de la notion de point-selle.

Initialisation : $k = 0$, choisir $\mu^{(0)} \in \mathbb{R}^p$ et $\lambda^{(0)} \in \mathbb{R}^{+m}$
Iteration. Tant que le critere d'arret n'est pas satisfait :
a) calculer $x^{(k)} \in \mathbb{R}^n$ solution de

$$\left(\mathcal{P}^{(k)}\right) \begin{cases} \min \ \mathscr{L}\left(x, \mu^{(k)}, \lambda^{(k)}\right) \\ x \in \mathbb{R}^n \end{cases}$$

b) calculer $\mu^{(k+1)}$ et $\lambda^{(k+1)}$ avec

$$\begin{cases} \mu_i^{(k+1)} = \mu_i^{(k)} + \rho \ f_i\left(x^{(k)}\right), i = 1, ..., p \\ \lambda_j^{(k+1)} = \max\left(0, \lambda_j^{(k)} + \rho g_j\left(x^{(k)}\right)\right), j = 1, ...m \end{cases}$$

ou $\rho > 0$, est un reel fixe (par l'utilisateur).

TAB. 5.6 – Algorithme d'Uzawa

**Définition 21** *On appelle point-selle de $\mathscr{L}$ sur $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+m}$ tout triplet $(x^*, \mu^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+m}$ vérifiant l'équation :*

$$\mathscr{L}(x^*, \mu, \lambda) \leqslant \mathscr{L}(x^*, \mu^*, \lambda^*) \leqslant \mathscr{L}(x, \mu^*, \lambda^*) \, \forall (x, \mu, \lambda) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+m}$$

On a alors le résultat suivant.

**Theorem 5.6** *Supposons que $J, f$ et $g$ soient des fonctions de classe $\mathcal{C}^1$ et que le triplet $(x^*, \mu^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+m}$ soit un point-selle de $\mathscr{L}$ sur $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+m}$. Alors, ce triplet vérifie les conditions de Kuhn-Tucker.*

Dans le cas important convexe, nous avons une caractérisation des points-selles grâce aux conditions de Kuhn-Tucker.

**Theorem 5.7** *Supponsons que $J, f$ et $g$ soient convexes et $\mathcal{C}^1$. Alors, le triplet $(x^*, \mu^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+m}$ est point-selle de $\mathscr{L}$ sur $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+m}$ si et seulement si il vérifie les conditions de Kuhn-Tucker.*

Le théorème précédent indique que nous allons chercher un triplet $(x^*, \mu^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+m}$ vérifiant les conditions de Kuhn-Tucker de la façon suivante :

1. Pour $(\mu^*, \lambda^*)$ fixés dans $\mathbb{R}^n \times (\mathbb{R}^+)^m$, nous allons chercher le minimum sans contrainte (sur tout $\mathbb{R}^n$) de la fonction $x \longmapsto \mathscr{L}(x, \mu^*, \lambda^*)$.
2. Pour $x^*$ fixé dans $\mathbb{R}^n$, on cherche le maximum sur $\mathbb{R}^p \times \mathbb{R}^{+m}$ (c'est-à-dire des contraintes de bornes simples) de la fonction $(\mu, \lambda) \longmapsto \mathscr{L}(x^*, \mu, \lambda)$

On fait ces deux calculs simultanément. On obtient alors l'algorithme d'Uzawa.

L'étape a) revient à résoudre :

$$\nabla_x \mathscr{L}\left(x, \mu^{(k)}, \lambda^{(k)}\right) = \nabla J\left(x\right) + \sum_{j=1}^{p} \mu_j^{(k)} \nabla f_j\left(x\right) + \sum_{i=1}^{m} \lambda_i^{(k)} \nabla g_i\left(x\right) = 0$$

La seconde étape est immédiate.

On a alors le théorème de convergence suivant.

**Theorem 5.8** *On suppose que $J$ est $\mathcal{C}^1$ et elliptique, que $f$ est affine, $g$ convexe de classe $\mathcal{C}^1$ et que $h$ et $g$ sont lipschitziennes.*

*On suppose de plus que le Lagrangien $\mathscr{L}$ possède un point-selle $(x^*, \mu^*, \lambda^*)$ sur $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^{+m}$.*

*Alors, il existe $\rho_1$, $\rho_2$, avec $0 < \rho_1 < \rho_2$ tels que $\forall \rho \in [\rho_1, \rho_2]$, la suite $\left(x^{(k)}\right)_{k>0}$ générée par l'algorithme d'Uzawz converge vers $x^*$.*

### 5.5.3 Méthode de programmation quadratique successive (SQP) (Sequential Quadratic Pragramming)

Nous allons nous attaquer maintenant à une dernière classe de méthodes appelées SQP pour Sequential Quadratic Programming.

Intéressons-nous dans un premier temps au cas de contraintes en égalité. Considérons le problème

$$\begin{cases} \min\ J\left(x\right) \\ x \in C. \end{cases}$$

où

$$C = \{x \in \mathbb{R}^n / f_i\left(x\right) = 0,\ i = 1, ..., p\}$$

Nous avons vu précédemment qu'une solution $x^*$ de ce problème est un point critique du Lagrangien $\mathcal{L}$ mais ce n'est pas en général un minimum de ce Lagrangien. Nous allons développer une méthode de descente particulière sur les conditions d'optimalité du système précédent. L'idée essentielle consiste à résoudre une succession de problèmes quadratiques avec contraintes linéaires (ces problèmes sont relativement simples à résoudre) qui sont des approximations du problème de départ.

Etant donné $x^{(k)}$, on cherche $x^{(k+1)} = x^{(k)} + \rho^{(k)} d^{(k)}$

où $d^{(k)} \in \mathbb{R}^n$ est une direction de descente et $\rho^{(k)} > 0$ le pas.

Effectuons une approximation des contraintes $f$ à l'aide de la formule de Taylor du premier ordre.

$$f_i\left(x^{(k)} + d\right) = f_i\left(x^{(k)}\right) + \nabla f_i\left(x^{(k)}\right) \cdot d + \mathcal{O}\left(\|d\|^2\right)$$

Si on néglige les termes d'ordre supérieur ou égal à 2, on définit la direction $d^{(k)}$ avec la direction permettant d'assumer $f_i\left(x^{(k)}+d\right)\simeq 0$.

On pose donc :

$$f_i\left(x^{(k)}\right)+\nabla f_i\left(x^{(k)}\right)\cdot d^{(k)}=0,\forall i=1,...,q.$$

ou encore :

$$Df\left(x^{(k)}\right)d^{(k)}=-f\left(x^{(k)}\right)$$

où $Df\left(x^{(k)}\right)$ est la matrice jacobienne de $f$ en $x^{(k)}$. Cette relation correspond à une linéarisation des contraintes au voisinage de $x^{(k)}$ : c'est un système linéaire.

Par ailleurs, il faudrait que $x^{(k+1)}$ diminue la valeur du Lagrangien (puisque c'est le Lagrangien qui joue le rôle de la fonction objectif quand ànd on a des contraintes). On va faire une approximation du Lagrangien

$$\mathscr{L}\left(x^{(k)}+\lambda,\mu\right)=\mathscr{L}\left(x^{(k)},\mu\right)+\left(\nabla_x\mathscr{L}\left(x^{(k)},\mu\right),\lambda\right)+\frac{1}{2}\left(D^2_{xx}\mathcal{L}\left(x^{(k)},\mu\right)d,d\right)+\mathcal{O}\left(\|d\|^3\right)$$

Si on néglige les termes d'ordre supérieur ou égal à 3, on voit qu'il faut minimiser

$$\left(\nabla_x\mathscr{L}\left(x^{(k)},\mu\right),\lambda\right)+\frac{1}{2}\left(D^2_{xx}\mathscr{L}\left(x^{(k)},\mu\right)d,d\right)$$

pour espérer minimiser le Lagrangien. On cherche donc, en fin de compte $d^{(k)}$ comme solution du problème

$$(QP)_e\begin{cases}\min\left(\nabla J\left(x^{(k)}\right)d\right)+\left(D^2_{xx}\mathcal{L}\left(x^{(k)},\mu\right)d,d\right)\\ Df\left(x^{(k)}\right)d^{(k)}+f\left(x^{(k)}\right)=0\end{cases}$$

En effet, nous avons :

$$\begin{aligned}\left(\nabla_x\mathscr{L}\left(x^{(k)},\mu\right),d\right) &= \left(\nabla J\left(x^{(k)}\right)d\right)+\mu^tDf\left(x^{(k)}\right)d\\ &= \left(\nabla J\left(x^{(k)}\right),d\right)+\mu^tf\left(x^{(k)}\right)\end{aligned}$$

Le dernier terme étant constant. Il reste ensuite à déterminer le pas $\rho^{(k)}$ et le multiplicateur $\mu^{(k)}$ à chaque itération. Il y a bien sûr, beaucoup de possibilités qui génèrent autant de variantes de la méthode.

Nous présentons ici, la méthode qui est basée sur le choix : $\rho^{(k)}=1$

Intéressons-nous maintenant au cas de contraintes générales en égalité et inégalité ; globalement, le principe est le même, seule la fonction Lagrangienne est modifiée.

Pour le problème :

$$(\mathcal{P})\begin{cases}\min\ J(x)\\ x\in C\end{cases}$$

Initialisation
$k = 1$, choix de $x^{(0)} \in \mathbb{R}$, $\mu^{(0)} \in \mathbb{R}^p$
Iteration $k$ : tant que le critere d'arret n'est pas satisfait
a) Resoudre le sous probleme quadratique :

$$(QP)_e \begin{cases} \min \left(\nabla J\left(x^{(k)}\right) d\right) + \frac{1}{2}\left(D^2_{xx}\mathcal{L}\left(x^{(k)}, \mu^{(k)}\right) d, d\right) \\ D\left(x^{(k)}\right) d + f\left(x^{(k)}\right) = 0 \end{cases}$$

b) on pose alors $\mu^{(k+1)} \in \mathbb{R}^p$ le multiplicateur associe a la contrainte (en egalite) de $(QP)_e$ et $x^{(k+1)} = x^{(k)} + d^{(k)}$
$k \longleftarrow k + 1$

TAB. 5.7 – Méthode SQP pour des contraintes en égalité

Initialisation
$k = 1$
Choix de $x^{(0)} \in \mathbb{R}^n, \left(\mu^{(0)}, \lambda^{(0)}\right) \in \mathbb{R}^p \times \mathbb{R}^{+m}$
Iteration $k$ : faire tant que le critere n'est pas satisfait
a) Resoudre le sous probleme quadratique

$$(QP) \begin{cases} \min \left(\nabla J\left(x^{(k)}\right), \lambda\right) + \frac{1}{2} \\ Df\left(x^{(k)}\right) \lambda + f\left(x^{(k)}\right) = 0 \\ Dg\left(x^{(k)}\right) \lambda + g\left(x^{(k)}\right) \leqslant 0 \end{cases}$$

$\mu^{(k+1)} \in \mathbb{R}^p$ est le multiplicateur associe a la contrainte en egalite de $(QP)$ et $\lambda^{(k+1)} \in \mathbb{R}^{+m}$ le multiplicateur (positif) associe a la contrainte en inegalite
$x^{(k+1)} = x^{(k)} + \lambda^{(k)}$
$k \longleftarrow k + 1$

TAB. 5.8 – Algorithme $SQP$ **pour des contraintes générales**

où

$$C = \{x \in \mathbb{R}^n / f(x) = 0, g(x) \leqslant 0\}$$

$$f = (f_i)_{1 \leqslant i \leqslant p}, g = (g_j)_{1 \leqslant j \leqslant m},$$

elle vaut

$$\mathscr{L}(x, \mu, \lambda) = J(x) + \mu^t f(x) + \lambda^t g(x)$$

où

$$\mu \in \mathbb{R}^p \; et \; \lambda \in \mathbb{R}^{+m}$$

La méthode SQP s'écrit de la façon suivante ou linéarise les contrainte et on fait une approximation quadratique de $\mathscr{L}$. On obtient alors l'algorithme suivant.

Il existe finalement pour ce type d'algorithme, des résultats de convergence de l'algorithme.

## 5.6 Exercices

**Exercice 5.1**

Soit $\mathbf{b} \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$, $r > 0$, et $B$, une matrice symétrique définie positive d'ordre $n$. On appelle $C$, l'ensemble défini par : $C := \{\mathbf{X} \in \mathbb{R}^n :< B\mathbf{X}, \mathbf{X} >= r^2\}$.
On définit la fonctionnelle $F : \mathbb{R}^n \longrightarrow \mathbb{R}$, par : $F(\mathbf{X}) =< \mathbf{a}, \mathbf{X} > +\alpha$. On désigne alors par $(\mathcal{P})$, le problème suivant :

$$\begin{cases} \min F(\mathbf{X}) \\ \mathbf{X} \in C \end{cases}$$

1. Démontrer que le problème $(\mathcal{P})$ admet au moins une solution.
2. Résoudre alors le problème $(\mathcal{P})$ et donner une interprétation géométrique.

**Exercice 5.2**

On considère le problème :

$$(\mathcal{P}) \begin{cases} \min\left\{\frac{1}{2}(A\mathbf{x}, \mathbf{x}) - (\mathbf{b}, \mathbf{x})\right\} \\ x_1 \geq 1 \\ x_2 - 2x_3 = 1 \end{cases}, \text{ avec } A = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix} \text{ et } \mathbf{b} = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}.$$

1. Démontrer que le problème $(\mathcal{P})$ admet une solution. Est-elle unique ?
2. Résoudre complètement $(\mathcal{P})$.
3. Comparer cette solution avec la solution du problème sans contrainte.

**Exercice 5.3** (Utilisation du Lagrangien).

Soit $f$, la fonction de deux variables définie par :

$$f(x_1, x_2) = (x_1 - 1)^2 + x_2 - 2.$$

Après avoir montré l'existence d'une solution, résoudre le problème :

$$(\mathcal{P}) \begin{cases} \min f(x_1, x_2) \\ x_2 - x_1 - 1 = 0 \\ x_1 + x_2 - 4 \leq 0 \end{cases}.$$

**Exercice 5.4** (Etude de l'efficience d'un portefeuille d'actions).

On considère un portefeuille d'actions composé de $n \geq 3$ actions *à risque* $(a_1, ..., a_n)$. On note $x_i$, la proportion de l'action $a_i$ dans le portefeuille. Le vecteur $\mathbf{x} = (x_1, ..., x_n)^T$ représente donc la composition du

portefeuille. On désigne par $\mathbf{u}$, le vecteur de taille $n$, de coordonnées $u_i = 1,\ \forall i \in \{1,...,n\}$. Il est évident que $\mathbf{x}$ vérifie alors :

$$\sum_{i=1}^{n} x_i = 1 = (\mathbf{u}, \mathbf{x}),\ \text{et}\ x_i \geq 0,\ \forall i \in \{1,...,n\}.$$

Le rendement de l'action $a_i$ est modélisé par une variable aléatoire $R_i$ de moyenne $e_i = \mathbb{E}(R_i)$. On introduit le vecteur de rendement moyen $\mathbf{e} = (e_1,...,e_n)^T$, puis la matrice de covariance $A = (a_{i,j})_{\substack{1\leq i\leq n\\ 1\leq j\leq n}}$ définie par la relation :

$$a_{i,j} = \mathbb{E}\left[(R_i - \mathbb{E}(R_i))\,(R_j - \mathbb{E}(R_j))\right], \forall (i,j) \in \{1,...,n\}^2.$$

La matrice $A$ est symétrique et positive. On suppose, de plus, que $A$ est définie.
Le rendement du portefeuille est calculé par la fonctionnelle $\varepsilon(\mathbf{x}) = (\mathbf{e}, \mathbf{x})$, avec $\mathbf{e} \in \mathbb{R}^n$ tel que $0 < e_1 < e_2 < ... < e_n$, tandis que le risque du portefeuille est calculé par la fonctionnelle $\sigma(\mathbf{x}) = \frac{1}{2}(A\mathbf{x}, \mathbf{x})$. On dit qu'un portefeuille $\mathbf{x}$ est *efficient* s'il assure à la fois un rendement maximal $\varepsilon$ pour un risque donné $\sigma$ et un risque minimal $\sigma$ pour un rendement imposé $\varepsilon$.

Pour $\varepsilon \in \mathbb{R}$ et $\sigma \in \mathbb{R}_+$ donnés, on définit les ensembles :

$$\begin{aligned} C_1(\varepsilon) &= \left\{\mathbf{x} \in \mathbb{R}^n : (\mathbf{u}, \mathbf{x}) = 1 \text{ et } (\mathbf{e}, \mathbf{x}) = \varepsilon\right\}, \\ C_2(\sigma) &= \left\{\mathbf{x} \in \mathbb{R}^n : (\mathbf{u}, \mathbf{x}) = 1 \text{ et } \frac{1}{2}(A\mathbf{x}, \mathbf{x}) = \sigma\right\}. \end{aligned}$$

Dans toute la suite de l'exercice, on cherche à résoudre les problèmes :

$$(\mathcal{P}_-)\begin{cases} \inf\left\{\frac{1}{2}(A\mathbf{x}, \mathbf{x})\right\} \\ \mathbf{x} \in C_1(\varepsilon) \end{cases} \quad \text{et} \quad (\mathcal{P}_+)\begin{cases} \sup(\mathbf{e}, \mathbf{x}) \\ \mathbf{x} \in C_2(\sigma) \end{cases}.$$

1. Montrer que pour tout $\varepsilon \in \mathbb{R}$, l'ensemble des contraintes $C_1(\varepsilon)$ est non vide, fermé et non borné.
   En déduire que le problème $(\mathcal{P}_-)$ admet une solution unique.

2. Montrer que pour certaines valeurs de $\sigma \in \mathbb{R}_+$ que l'on précisera, l'ensemble $C_2(\sigma)$ est non vide, fermé et borné.
   En déduire que le problème $(\mathcal{P}_+)$ admet au moins une solution.

3. On appelle $\lambda$, le multiplicateur de Lagrange associé à la contrainte $(\mathbf{u}, \mathbf{x}) = 1$, et $\mu$, le multiplicateur de Lagrange associé à la contrainte $(\mathbf{e}, \mathbf{x}) = \varepsilon$. On définit ensuite les réels $a$, $b$ et $c$ par :

   $$a = (A^{-1}\mathbf{u}, \mathbf{u}),\ b = (A^{-1}\mathbf{u}, \mathbf{e}),\ c = (A^{-1}\mathbf{e}, \mathbf{e}) \text{ et } d = b^2 - ac.$$

   En utilisant les conditions d'optimalité associées à l'ensemble $C_1(\varepsilon)$, montrer que le solution $x$ du

problème $(\mathcal{P}_-)$ vérifie :

$$\lambda = \frac{c - b\varepsilon}{d}; \tag{5.1}$$

$$\mu = \frac{a\varepsilon - b}{d}; \tag{5.2}$$

$$\mathbf{x} = -A^{-1}(\lambda \mathbf{u} + \mu \mathbf{e}). \tag{5.3}$$

4. On dit qu'une solution $\mathbf{x}$ est *efficiente* si elle est solution commune aux problèmes $(\mathcal{P}_-)$ et $(\mathcal{P}_+)$, et on appelle **frontière d'efficience**, la courbe, dans le plan des $(\varepsilon, \sigma)$ correspondant à l'ensemble de ces solutions quand $\varepsilon$ et $\sigma$ varient.
   Déterminer la frontière d'efficience des problèmes $(\mathcal{P}_-)$ et $(\mathcal{P}_+)$.

**Exercice 5.5** (Méthode de pénalisation).

Soient $J : \mathbb{R}^n \longrightarrow \mathbb{R}$, une fonction strictement convexe et coercive. Soient $m$, un entier naturel non nul, et pour tout $i \in \{1, ..., m\}$, $\varphi_i : \mathbb{R}^n \longrightarrow \mathbb{R}$, des fonctions convexes. On désigne par $C$ l'ensemble :

$$C := \{\mathbf{X} \in \mathbb{R}^n : \varphi_i(\mathbf{X}) \leq 0, 1 \leq i \leq m\}.$$

Supposons que $C$ est borné. On note $\mathring{C}$, son intérieur défini par :

$$\mathring{C} = \{\mathbf{X} \in \mathbb{R}^n : \varphi_i(\mathbf{X}) < 0, 1 \leq i \leq m\}.$$

Pour $\varepsilon > 0$, on définit la fonctionnelle **pénalisée** $J_\varepsilon$ par :

$$J_\varepsilon(\mathbf{X}) = J(\mathbf{X}) - \varepsilon \sum_{i=1}^{m} \frac{1}{\varphi_i(\mathbf{X})}, \ \mathbf{X} \in \mathring{C}.$$

1. Montrer que $J_\varepsilon$ possède sur $\mathring{C}$ un unique minimum que l'on notera $\mathbf{X}_\varepsilon$. Pour l'existence du minimum, on pensera à introduire une suite minimisante.
2. On souhaite minimiser la fonctionnelle $J$ sur l'ensemble des contraintes $C$.
   (a) Montrer que, quitte à extraire, $\mathbf{X}_\varepsilon \xrightarrow[\varepsilon \to 0]{} \mathbf{X}^\star$.
   (b) Montrer que $\mathbf{X}^\star$ est la solution du problème de minimisation.
3. Montrer que : $0 < \varepsilon < \varepsilon' \Longrightarrow J(\mathbf{X}^\star) \leq J(\mathbf{X}_\varepsilon) \leq J(\mathbf{X}_{\varepsilon'})$.

**Exercice 5.6** (Algorithmes d'Uzawa et d'Arrow-Hurwicz).

Soit $A$, une matrice symétrique et définie positive, et $\mathbf{b}$, un vecteur donné de $\mathbb{R}^n$. On définit la fonctionnelle quadratique $J$ pour $\boldsymbol{\nu} \in \mathbb{R}^n$ par :

$$J(\boldsymbol{\nu}) = \frac{1}{2}(A\boldsymbol{\nu}, \boldsymbol{\nu}) - (\mathbf{b}, \boldsymbol{\nu}).$$

Soit $C \in \mathcal{M}^{m\times n}(\mathbb{R})$. On appelle $(\mathcal{P})$ le problème :

$$\begin{cases} \inf J(\boldsymbol{\nu}) \\ \boldsymbol{\nu} \in U = \{\boldsymbol{\nu} \in \mathbb{R}^n : C\boldsymbol{\nu} = \mathbf{0}\}. \end{cases}$$

1. Montrer que le problème $(\mathcal{P})$ admet une unique solution $\mathbf{u}$ telle que $J(\mathbf{u}) = \inf_{\boldsymbol{\nu}\in U} J(\boldsymbol{\nu})$.
2. Écrire l'algorithme si l'on souhaite résoudre ce problème en utilisant la méthode d'Uzawa.
3. Soient $\rho_1$ et $\rho_2$, deux paramètres strictement positifs. On définit la méthode itérative suivante :

$$\begin{cases} (\mathbf{u}^0, \boldsymbol{\lambda}^0) \text{ donnés dans } \mathbb{R}^n \times \mathbb{R}^m \\ (\mathbf{u}^k, \boldsymbol{\lambda}^k) \text{ étant connus, calcul de } (\mathbf{u}^{k+1}, \boldsymbol{\lambda}^{k+1}) \ : \\ \quad \bullet\ \mathbf{u}^{k+1} = \mathbf{u}^k - \rho_1(A\mathbf{u}^k - \mathbf{b} + C^T\boldsymbol{\lambda}^k)\ ; \\ \quad \bullet\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho_1\rho_2 C\mathbf{u}^{k+1}\ ; \end{cases} \tag{5.4}$$

   On définit le nombre réel $\beta$ par : $\beta = \|I - \rho_1 A\|$.
   Démontrer que, si $\rho_1$ est suffisamment petit, alors $\beta < 1$.
4. (a) Soit $\boldsymbol{\lambda}$, un vecteur de $\mathbb{R}^m$ qui vérifie : $A\mathbf{u} + C^T\boldsymbol{\lambda} = \mathbf{b}$.
   Expliquer pourquoi il existe de tels vecteurs.
   (b) On choisit à présent le paramètre $\rho_1$ pour que l'inégalité $\beta < 1$ ait lieu. Montrer que, si le paramètre $\rho_2 > 0$ est suffisamment petit, il existe une constante $\gamma > 0$, indépendante de l'entier $k$, telle que :

$$\gamma \left\| \mathbf{u}^{k+1} - \mathbf{u} \right\|_n^2 \leq \left( \frac{\left\| \boldsymbol{\lambda}^k - \boldsymbol{\lambda} \right\|_m^2}{\rho_2} - \beta \left\| \mathbf{u}^k - \mathbf{u} \right\|_n^2 \right) + \left( \frac{\left\| \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda} \right\|_m^2}{\rho_2} + \beta \left\| \mathbf{u}^{k+1} - \mathbf{u} \right\|_n^2 \right).$$

   (c) En déduire que, pour de tels choix des paramètres $\rho_1$ et $\rho_2$, on a : $\lim_{k\to+\infty}(\mathbf{u}_k) = \mathbf{u}$.
5. Que peut-on dire de la suite $(\boldsymbol{\lambda}^k)$ lorsque le rang de la matrice $C$ est $m$ ?
6. L'algorithme ci-dessus s'appelle **l'algorithme d'Arrow-Hurwicz**.
   Quel avantage présente cet algorithme par rapport à la méthode d'Uzawa ?

**Exercice 5.7**

**Rappel de Géométrie :** *un cône de sommet $\Omega$ est formée d'une famille de droites passant par $\Omega$. Ces droites sont les génératrices du cône. Une courbe qui rencontre toutes les génératrices est une directrice.*

On se place dans $\mathbb{R}^3$. On appelle $\mathcal{C}$, le cône de révolution de sommet $\Omega(0,-1,2)$ et de directrice le cercle $\Gamma$ du plan $\{y = 0\}$ de centre $(0,0,2)$ et de rayon 1. On considère le demi espace $D_\mu$ d'équation $-y + \mu \leq 0$, avec $\mu \geq -1$. On désigne par $U$, la région convexe intersection de l'intérieur de $\mathcal{C}$ avec $D_\mu$.

1. Faire un dessin et écrire l'équation de $\mathcal{C}$.
2. Déterminer, en discutant suivant les valeurs de $\mu$, le point de $\overline{U}$ réalisant le minimum de la distance de $O$ (l'origine) à ce convexe fermé.
   **Remarque :** on n'oubliera pas de justifier l'existence du minimum.

**Exercice 5.8**

Cet exercice a pour but l'étude de la convergence de méthodes itératives analogues à la méthode d'Uzawa. Soit $n$ un entier non nul et $U_0$, un sous ensemble de $\mathbb{R}^n$. On note $< .,. >_n$, le produit scalaire usuel de $\mathbb{R}^n$, $\|.\|_n$, la norme associée. Soit $J : \mathbb{R}_n \longrightarrow \mathbb{R}$, une fonctionnelle, $\phi_1$, ..., $\phi_m$, $m$ fonctions de $\mathbb{R}^n$ dans $\mathbb{R}$.
On fait **une fois pour toutes** les hypothèses suivantes :

($i$) $J$ est elliptique, i.e. elle est une fois continûment dérivable dans $\mathbb{R}^n$ et il existe une constante $\alpha > 0$ telle que :

$$< \nabla J(\mathbf{v}) - \nabla J(\mathbf{u}), \mathbf{v} - \mathbf{u} >_n \geq \alpha \|\mathbf{v} - \mathbf{u}\|_n^2, \ \forall (\mathbf{u}, \mathbf{v}) \in (\mathbb{R}^n)^2.$$

($ii$) Il existe une constante $M > 0$ telle que :

$$\|\nabla J(\mathbf{v}) - \nabla J(\mathbf{u})\|_n \leq M \|\mathbf{v} - \mathbf{u}\|_n, \ \forall (\mathbf{u}, \mathbf{v}) \in (\mathbb{R}^n)^2.$$

($iii$) L'ensemble $U_0$ est un convexe fermé de $\mathbb{R}^n$.
($iv$) L'ensemble $U$ est non vide.
($v$) Les fonctions $\{\phi_i\}_{1 \leq i \leq m}$ sont convexes.
($vi$) Il existe une constante $C > 0$ telle que :

$$\|\boldsymbol{\phi}(\mathbf{v}) - \boldsymbol{\phi}(\mathbf{u})\|_m \leq M \|\mathbf{v} - \mathbf{u}\|_n, \ \forall (\mathbf{u}, \mathbf{v}) \in (\mathbb{R}^n)^2,$$

où $\boldsymbol{\phi} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ désigne l'application dont les composantes sont les fonctions $\phi_i$.
On considère le problème :

$$\text{Trouver } \mathbf{u} \in \mathbb{R}^n \text{ tel que } (\mathcal{P}) \ \begin{cases} \mathbf{u} \in U := U_0 \cap \{\mathbf{v} \in \mathbb{R}^n : \phi_i(\mathbf{v}) \leq 0, \ 1 \leq i \leq m\} \\ J(\mathbf{u}) = \inf_{\mathbf{v} \in U} J(\mathbf{v}). \end{cases}$$

## Partie I : étude théorique

1. Établir les inégalités :

$$< \nabla J(\mathbf{u}), \mathbf{v} - \mathbf{u} >_n + \frac{\alpha}{2} \|\mathbf{v} - \mathbf{u}\|_n^2 \leq J(\mathbf{v}) - J(\mathbf{u}) \leq < \nabla J(\mathbf{u}), \mathbf{v} - \mathbf{u} >_n + \frac{M}{2} \|\mathbf{v} - \mathbf{u}\|_n^2, \ \forall (\mathbf{u}, \mathbf{v}) \in (\mathbb{R}^n)^2.$$

2. Démontrer que le problème $(\mathcal{P})$ possède une solution et une seule.

3. On définit le Lagrangien associé au problème $(\mathcal{P})$ comme étant la fonction :

$$\begin{array}{rccl} \mathcal{L} : & U_0 \times \mathbb{R}_+^m & \longrightarrow & \mathbb{R} \\ & (\mathbf{v}, \boldsymbol{\mu}) & \longmapsto & J(\mathbf{v}) + < \boldsymbol{\mu}, \boldsymbol{\phi}(\mathbf{v}) >_m . \end{array}$$

Démontrer que si $(\mathbf{u}, \boldsymbol{\lambda}) \in U_0 \times \mathbb{R}_+^m$ est un point-selle du Lagrangien $\mathcal{L}$ sur l'ensemble $U_0 \times \mathbb{R}_+^m$, alors le point $\mathbf{u}$ est solution du problème $(\mathcal{P})$.

4. On appelle $\Pi_+$, l'opérateur de projection de $\mathbb{R}^m$ sur $\mathbb{R}_+^m$. Soit $\rho > 0$ fixé.

Vérifier l'équivalence :

$$\boldsymbol{\lambda} = \Pi_+ \left(\boldsymbol{\lambda} + \rho\boldsymbol{\phi}(\mathbf{u})\right) \Longleftrightarrow \left\{ \begin{array}{l} \boldsymbol{\lambda} \in \mathbb{R}^m_+,\ \boldsymbol{\phi}(\mathbf{u}) \leq 0\ ; \\ < \boldsymbol{\lambda}, \boldsymbol{\phi}(\mathbf{u}) >_m = 0. \end{array} \right.$$

5. Vérifier qu'un couple $(\mathbf{u}, \boldsymbol{\lambda})$ est un point-selle du Lagrangien $\mathcal{L}$ si, et seulement si :

$$\left\{ \begin{array}{l} < \nabla J(\mathbf{u}), \mathbf{v} - \mathbf{u} >_n + < \boldsymbol{\lambda}, \boldsymbol{\phi}(\mathbf{v}) >_m \geq 0,\ \forall \mathbf{v} \in U_0\ ; \\ \boldsymbol{\lambda} = \Pi_+ \left(\boldsymbol{\lambda} + \rho\boldsymbol{\phi}(\mathbf{u})\right),\ \rho > 0 \text{ fixé mais arbitraire.} \end{array} \right.$$

## Partie II : étude numérique

Soient $\varepsilon$ et $\rho$, deux nombres réels strictement négatifs fixés. On définit une méthode itérative de la façon suivante :

- On part d'un couple $(\mathbf{u}^0, \boldsymbol{\lambda}^0) \in U_0 \times \mathbb{R}^m_+$ arbitraire.
- On définit alors pour $k \geq 0$ une suite de couples $(\mathbf{u}^k, \boldsymbol{\lambda}^k) \in U_0 \times \mathbb{R}^m_+$ par récurrence :
  - **Calcul de $\mathbf{u}^{k+1}$ :** $\mathbf{u}^{k+1} \in U_0$, et vérifie :

$$f(\mathbf{u}^{k+1}) = \inf_{\mathbf{v} \in U_0} \{f(\mathbf{v})\}, \text{ où } f : \begin{array}{ccc} \mathbb{R}^n & \longrightarrow & \mathbb{R} \\ \mathbf{v} & \longmapsto & \frac{1}{2}\|\mathbf{v}\|^2_n + \left\langle \varepsilon \nabla J(\mathbf{u}^k) - \mathbf{u}^k, \mathbf{v} \right\rangle_n + \varepsilon \left\langle \boldsymbol{\lambda}^k, \boldsymbol{\phi}(\mathbf{v}) \right\rangle_m \end{array}.$$

  - **Calcul de $\boldsymbol{\lambda}^{k+1}$ :** $\boldsymbol{\lambda}^{k+1} = \Pi_+ \left(\boldsymbol{\lambda}^k + \rho\boldsymbol{\phi}(\mathbf{u}^{k+1})\right)$.

1. Démontrer que le problème de minimisation définissant le vecteur $\mathbf{u}^{k+1}$ à partir du couple $(\mathbf{u}^k, \boldsymbol{\lambda}^k)$ admet une solution et une seule.
2. Démontrer alors que le vecteur $\mathbf{u}^{k+1}$ est solution de ce problème si, et seulement si :

$$\mathbf{u}^{k+1} \in U_0 \text{ et } \left\langle \mathbf{u}^{k+1} - \mathbf{u}^k + \varepsilon \nabla J(\mathbf{u}^k), \mathbf{v} - \mathbf{u}^{k+1} \right\rangle_n + \varepsilon \left\langle \boldsymbol{\lambda}^k, \boldsymbol{\phi}(\mathbf{v}) - \boldsymbol{\phi}(\mathbf{u}^{k+1}) \right\rangle_m \geq 0,\ \forall \mathbf{v} \in U_0.$$

3. Soit $(\mathbf{u}, \boldsymbol{\lambda})$, un point-selle du Lagrangien $\mathcal{L}$. Établir les inégalités suivantes :

$$(i) \quad 2 \left\langle \boldsymbol{\lambda}^k - \boldsymbol{\lambda}, \boldsymbol{\phi}(\mathbf{u}) - \boldsymbol{\phi}(\mathbf{u}^{k+1}) \right\rangle_m \leq \frac{1}{\rho} \left( \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}\|^2_m - \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}\|^2_m \right) + \rho C^2 \|\mathbf{u}^{k+1} - \mathbf{u}\|^2_n\ ;$$

$$(ii) \quad \left\langle \mathbf{u}^{k+1} - \mathbf{u}^k, \mathbf{u} - \mathbf{u}^{k+1} \right\rangle_n + \varepsilon \left\langle \nabla J(\mathbf{u}^k) - \nabla J(\mathbf{u}), \mathbf{u} - \mathbf{u}^{k+1} \right\rangle_n + \varepsilon \left\langle \boldsymbol{\lambda}^k - \boldsymbol{\lambda}, \boldsymbol{\phi}(\mathbf{u}) - \boldsymbol{\phi}(\mathbf{u}^{k+1}) \right\rangle_m \geq 0\ ;$$

$$(iii) \quad \left\langle \nabla J(\mathbf{u}^k) - \nabla J(\mathbf{u}), \mathbf{u} - \mathbf{u}^{k+1} \right\rangle_n \leq \frac{M}{2} \|\mathbf{u}^k - \mathbf{u}^{k+1}\|^2_n - \frac{\alpha}{2} \left( \|\mathbf{u}^k - \mathbf{u}\|^2_n + \|\mathbf{u}^{k+1} - \mathbf{u}\|^2_n \right)\ ;$$

$$(iv) \quad \frac{1}{2}\|\mathbf{u}^{k+1}\|^2_n + \left\langle \mathbf{u}^{k+1}, \mathbf{u} - \mathbf{u}^{k+1} \right\rangle_n - \frac{1}{2}\|\mathbf{u}^k\|^2_n + \left\langle \mathbf{u}^k, \mathbf{u} - \mathbf{u}^k \right\rangle_n + \frac{1}{2}(\varepsilon M - 1)\|\mathbf{u}^k - \mathbf{u}^{k+1}\|^2_n + \varepsilon \frac{\alpha}{2} \left( \|\mathbf{u}^{k+1} - \mathbf{u}\|^2_n - \|\mathbf{u}^k - \mathbf{u}\|^2_n \right) + \varepsilon \left( \rho \frac{C^2}{2} - \alpha \right) \|\mathbf{u}^{k+1} - \mathbf{u}\|^2_n + \frac{\varepsilon}{2\rho} \left( \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}\|^2_m - \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}\|^2_m \right) \geq 0.$$

4. Déduire de la dernière inégalité de la question précédente que si $0 < \varepsilon \leq \dfrac{1}{M}$, et $0 < \rho < \dfrac{2\alpha}{C^2}$, alors $\lim_{k \to +\infty} (\mathbf{u}^k) = \mathbf{u}$ et que la suite $(\boldsymbol{\lambda}^k)_{k \geq 0}$ est bornée.

**Exercice 5.9**

On souhaite maximiser dans $\mathbb{R}^2$ la fonction :

$$\begin{array}{rccc} f : & \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\ & (x,y) & \longmapsto & 2x - x^2 + y. \end{array}$$

sur l'ensemble $K$ défini par :

$$K = \{(x,y) \in \mathbb{R}^2 : x + y \leq 1 \text{ et } xy \geq 0\}.$$

On appelle $g$, la fonctionelle définie sur $\mathbb{R}^2$ par : $g(x,y) = -f(x,y)$.

1. Démontrer l'existence de solutions pour ce problème d'optimisation.
2. **Résolution graphique du problème.**
   (a) Quelle est, dans le plan $xOy$, la ligne de niveau $\alpha$ de la fonctionnelle $g$ ? (autrement dit, l'ensemble des couples $(x,y)$ de $\mathbb{R}^2$ tels que $g(x,y) = \alpha$)
   (b) En représentant quelques lignes de niveau sur un graphique ainsi que l'ensemble des contraintes $K$, déterminer, à l'aide d'une méthode géométrique élémentaire que l'on précisera, la ou les solution(s) du problème ci-dessus.
3. Écrire les conditions d'optimalité pour ce problème puis le résoudre. On verifiera que l'on obtient la même solution que dans la question précédente.
4. Est-il possible de programmer en Matlab© un algorithme de résolution d'un tel problème à l'aide de la méthode de projection ? Expliquez.
   Proposer une solution acceptable, compte tenu de la localisation des solutions dans le plan.

**Exercice 5.10**

Soient $A$, une matrice carrée symétrique définie positive de taille $n$, $b$ et $c$, deux vecteurs non nuls de $\mathbb{R}^n$, et $\alpha \in \mathbb{R}$. On définit la fonctionnelle $F$ par la relation :

$$F(X) = \frac{1}{2} < A\mathbf{X}, \mathbf{X} > - < \mathbf{b}, \mathbf{X} > .$$

On définit également un ensemble de contraintes $C$ par la relation :

$$C = \{\mathbf{X} \in \mathbb{R}^n :< \mathbf{c}, \mathbf{X} >= \alpha\}.$$

On considère ici la méthode de dualité pour chercher un minimum de $F$ sur $C$.

1. Définir le Lagrangien $\mathcal{L}$ associé à ce problème. On précisera son ensemble de définition.

2. Soient $\lambda_0$ et $\rho$, deux nombres positifs donnés.
   Écrire soigneusement l'algorithme d'Uzawa $(\mathbf{X}_k, \lambda_k)$ en donnant les valeurs explicites de $\mathbf{X}_k$ et $\lambda_k$ en fonction des termes d'ordre $k-1$.
3. (a) Démontrer l'existence d'un valeur de $\rho$ que l'on notera $\rho_0$ pour laquelle la suite $(\lambda_k)_{k\in\mathbb{N}}$ est stationnaire à partir du rang 1.
   (b) Montrer qu'alors, $(\mathbf{X}_k)_{k\in\mathbb{N}}$ prend aussi une valeur constante. Quelle est cette valeur ?
4. Montrer que si $\rho > \rho_0$, alors les suites $(\mathbf{X}_k)_{k\in\mathbb{N}}$ et $(\lambda_k)_{k\in\mathbb{N}}$ sont convergentes.
   Déterminer leurs limites.

## 5.7 Travaux pratiques

### 5.7.1 Travaux pratiques 1

*L'objectif de cette séance de travaux pratiques est de vous apprendre à coder des algorithmes généraux de minimisation avec contraintes : l'algorithme d'Uzawa et la méthode de pénalisation. L'algorithme d'Uzawa est rappelé au début du TP.*

# Partie I : Quelques rappels de cours

**I.1 L'algorithme du gradient projeté.**

La méthode du gradient projeté s'inspire des méthodes usuelles de gradient. Supposons, d'une façon générale, que l'on souhaite minimiser une fonctionnelle $J : \mathbb{R}^n \longrightarrow \mathbb{R}$ sur un ensemble de contraintes $C$. Si l'on construit une suite d'itérés de la forme $\mathbf{x}^{k+1} = \mathbf{x}^k + \rho_k \mathbf{d}^k$, où $\mathbf{d}^k$ est une direction de descente, on ne peut pas être sûr que si $\mathbf{x}^k$ appartient à $C$, alors $\mathbf{x}^{k+1}$ appartiendra encore à $C$. Il faut donc "ramener" $\mathbf{x}^{k+1}$ dans $C$, ce que l'on fait en utilisant une projection.

**Algorithme du gradient projeté**

1. *Initialisation.*
   $k = 0$ : on choisit $\mathbf{x}^0 \in \mathbb{R}^n$ et $\rho_0 \in \mathbb{R}_+^*$.
2. *Itération $k$.*

$$\mathbf{x}^{k+1} = \Pi_C \left( \mathbf{x}^k - \rho_k \nabla J(\mathbf{x}^k) \right).$$

   $\Pi_C$ désigne ici la projection sur $C$

Notons également le résultat de convergence :

**Theorem 5.9** *On suppose que $J$ est $\mathcal{C}^1$, de dérivée Lipschitzienne, et elliptique, c'est à dire qu'il existe $\alpha > 0$ tel que :*

$$\forall (\mathbf{x}, \mathbf{y}) \in (\mathbb{R}^n)^2, \ (\nabla J(\mathbf{x}) - \nabla J(\mathbf{y}), \mathbf{x} - \mathbf{y}) \geq \alpha . \|\mathbf{x} - \mathbf{y}\|^2.$$

*Si l'on choisit le pas $\rho_k$ dans un intervalle $[\beta_1, \beta_2]$ tel que $0 < \beta_1 < \beta_2 < \frac{2\alpha}{M}$, où $\alpha$ est la constante d'ellipticité de $J$ et $M$, la constante de Lipschitz de la dérivée de $J$, alors la suite $(\mathbf{x}^n)_{n\geq 0}$ d'itérés par la méthode du gradient projeté converge vers la solution du problème de minimisation.*

### I.2 L'algorithme d'Uzawa.

Supposons que l'on souhaite résoudre sur $\mathbb{R}^n$ le problème $(\mathcal{P})$ $\min J(\mathbf{x}),\ \mathbf{x} \in C$, où :

$$C = \{\mathbf{x} \in \mathbb{R}^n, h(\mathbf{x}) = 0,\ g(\mathbf{x}) \leq 0\}, \text{ avec } \mathbf{h} = (h_i)_{1\leq i\leq p} \text{ et } \mathbf{g} = (g_j)_{1\leq j\leq q}.$$

Appelons $\mathcal{L}$, la fonction Lagrangienne associée à ce problème. Alors :

$$\forall \mathbf{x} \in \mathbb{R}^n,\ \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = J(\mathbf{x}) + (\boldsymbol{\lambda}, \mathbf{h}(\mathbf{x}))_{\mathbb{R}^p} + (\boldsymbol{\mu}, g(\mathbf{x}))_{\mathbb{R}^q}\,.$$

On voit ici que la fonction Lagrangienne englobe à la fois la fonctionnelle $J$ et les contraintes $\mathbf{h}$ et $\mathbf{g}$. Elle représente donc bien le problème $(\mathcal{P})$. Avant de poursuivre, souvenons-nous de ce que l'on appelle **point selle**.

**Définition 22** *On appelle **point selle** de $\mathcal{L}$ sur $\mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$, tout triplet $(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$ vérifiant l'équation :*

$$\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) \leq \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \leq \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*),\ \forall (\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q. \tag{5.5}$$

Le théorème suivant nous permettra de comprendre *intuitivement* l'algorithme d'Uzawa. Pour une compréhension totale, il faudra s'intéresser au problème dual de $(\mathcal{P})$.

**Theorem 5.10** *Supposons $J$, $g$ et $h$ convexes, de classe $\mathcal{C}^1$.*
*Alors, le triplet $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$ est un point selle de $\mathcal{L}$ si, et seulement si ce triplet vérifie les conditions de Kuhn-Tucker.*

Ce théorème nous aide à comprendre que, pour chercher le triplet $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$ vérifiant les conditions de Kuhn-Tucker, on peut procéder de la façon suivante :

- Pour $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in \mathbb{R}^p \times (\mathbb{R}_+)^q$, fixés, on peut chercher le minimum sans contrainte (i.e. sur tout l'espace $\mathbb{R}^n$) de la fonction $\mathbf{x} \longmapsto \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$. Cela traduit le terme de droite de l'équation (5.5).
- Pour $\mathbf{x}^* \in \mathbb{R}^n$ fixé, on cherche le maximum sur $\mathbb{R}^p \times (\mathbb{R}_+)^q$ de la fonction $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \longmapsto \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$. C'est ce que traduit le terme de gauche de l'équation (5.5).

C'est cette idée qu'utilise l'algorithme d'Uzawa :

**Algorithme d'Uzawa**

1. *Initialisation.*
   $k = 0$ : on choisit $\boldsymbol{\lambda}^0 \in \mathbb{R}^p$ et $\boldsymbol{\mu}^0 \in (\mathbb{R}_+)^q$.
2. *Itération $k$.*
   $\boldsymbol{\lambda}^k = (\lambda_1^k, ..., \lambda_p^k) \in \mathbb{R}^p$ et $\boldsymbol{\mu}^k = (\boldsymbol{\mu}_1^k, ..., \boldsymbol{\mu}_q^k) \in \mathbb{R}^q$ sont connus.
   (a) Calcul de $\mathbf{x}^k \in \mathbb{R}^n$ solution de :

$$(\mathcal{P}_k) \quad \min_{\mathbf{x} \in \mathbb{R}^n} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k).$$

   (b) Calcul de $\boldsymbol{\lambda}^{k+1}$ et $\boldsymbol{\mu}^{k+1}$ par les formules :

$$\begin{aligned} \lambda_i^{k+1} &= \lambda_i^k + \rho h_i(x^k), \ i \in \{1, ..., p\} \\ \mu_j^{k+1} &= \max(0, \mu_j^k + \rho g_j(x^k)), \ j \in \{1, ..., q\}. \end{aligned}$$

Enfin, signalons le théorème suivant qui pourrait se révéler utile en pratique :

**Theorem 5.11** *On suppose que $J$ est $\mathcal{C}^1$ et elliptique. Supposons de plus que* $\mathbf{h}$ *est affine,* $\mathbf{g}$ *est convexe de classe $\mathcal{C}^1$ et lipschitziennes. On suppose de plus que le Lagrangien $\mathcal{L}$ possède un point selle $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ sur $\mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$. Alors, il existe $\rho_1$ et $\rho_2$, avec $0 < \rho_1 < \rho_2$ tels que, pour tout $\rho \in [\rho_1, \rho_2]$, la suite $(\mathbf{x}^k)_{k \geq 0}$ générée par l'algorithme d'Uzawa converge vers $\mathbf{x}^*$.*
*De plus, on sait que $\rho_2 = \dfrac{\alpha}{2(M_h^2 + M_g^2)}$, avec $\alpha$, la constante d'ellipticité de $J$, $M_h$ et $M_g$, les constantes de Lipschitz associées à* $\mathbf{h}$ *et* $\mathbf{g}$.

# Partie II : Exercices

**Exercice 5.11** (Optimisation d'un portefeuille d'actions).

On considère le problème de l'Optimisation d'un portefeuille. Supposons que l'on possède $n$ actions, que l'on représente par des variable aléatoires $R_1$, ..., $R_n$. Chaque action rapporte en moyenne à l'actionnaire $e_i = \mathbb{E}(R_i)$ (espérance de $R_i$) au bout d'un an. On suppose que l'on investit une somme $S$ donnée, et l'on note $x_i \in \mathbb{R}$, la proportion de la somme investie dans l'action $i$. Ainsi, on a : $\sum_{i=1}^{n} x_i = 1$. Le portefeuille total est représenté par la variable aléatoire : $R = \sum_{i=1}^{n} x_i R_i$ et rapporte donc en moyenne : $\mathbb{E}(R) = \sum_{i=1}^{n} x_i e_i$.

On désire imposer un rendement donné $r_0 > 0$, ce qui se traduit par : $r_0 = \sum_{i=1}^{n} x_i e_i$.

On modélise le risque du portefeuille par : $\sigma^2(\mathbf{x}) = \mathbb{E}[(R - \mathbb{E}(R))^2]$.

On note $A = (a_{ij})_{\substack{1\leq i\leq n\\ 1\leq j\leq n}}$, la matrice de covariance définie par la relation :

$$\forall (i,j) \in \{1,...,n\}^2,\ a_{ij} = \mathbb{E}[(R_i - \mathbb{E}(R_i))(R_j - \mathbb{E}(R_j))].$$

On peut alors écrire que $\sigma^2(\mathbf{x}) = (\mathbf{x}, A\mathbf{x})$. On appelle $J$, la fonctionnelle définie sur $\mathbb{R}^n$ par :

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x}, A\mathbf{x}).$$

On appelle également $K$, l'ensemble des contraintes : $K := \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x}, \mathbf{u}) = 1 \text{ et } (\mathbf{x}, \mathbf{e}) = r_0\}$.
Le but de ce TP est de déterminer numériquement la solution du problème :

$$(\mathcal{P}) \begin{cases} \min J(\mathbf{x}) \\ \mathbf{x} \in K \end{cases}.$$

1. Mettre l'ensemble des contraintes sous la forme $K = \{\mathbf{x} \in \mathbb{R}^n : C\mathbf{x} = \mathbf{f}\}$, où $C$ et $\mathbf{f}$ désignent respectivement une matrice et un vecteur à préciser. Rappeler comment se traduisent les conditions d'Optimalité de ce problème et formuler l'équation en $\mathbf{x}$ à résoudre à chaque itération.

2. On souhaite étudier un exemple concret. Supposons que : $\forall i \in \{1,...,n\},\ e_i = i$, que $r_0 = 2.5$. Pour les tests numériques, on se placera par exemple dans le cas où $n = 5$.
   Écrire un programme *genere.m* permettant de générer la matrice $A$ à l'aide des instructions suivantes :

   ```
   A=diag(e./n);
   R=rand(n,n);
   A=A+0.1.*R'*R;
   ```

   Expliquer la dernière ligne du programme.

3. Pour différentes matrices $A$, programmer l'algorithme d'Uzawa. On n'oubliera pas d'imposer un nombre maximal d'itérations.

4. Quel inconvénient majeur constatez-vous ici ?

5. On appelle donc à présent $\tilde{K}$, l'ensemble défini par :

   $$\tilde{K} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq 0,\ (\mathbf{x}, \mathbf{u}) = 1 \text{ et } (\mathbf{x}, \mathbf{e}) = r_0\}.$$

   Si l'on souhaite améliorer la résolution du problème, on est amené à résoudre : $\min_{\mathbf{x}\in\tilde{K}} J(\mathbf{x})$.

6. Comment doit-on choisir la constante $\rho$ qui intervient dans l'algorithme d'Uzawa. Soyez précis.

7. Écrire l'algorithme d'Uzawa écrit sous sa forme la plus générale, puis le tester.

**Exercice 5.12** (Mise en œuvre d'une méthode de pénalisation).

On considère la fonctionnelle $J$ définie sur $\mathbb{R}^2$ par : $J(x,y) = 2x^2 + 3xy + 2y^2$.
On appelle $Q$ le cadrant défini par :

$$Q = \left\{ x \leq -\frac{1}{2},\ y \leq -\frac{1}{2} \right\}$$

1. (a) Quelle est la solution du problème de minimisation de $J$ sans contrainte ?
   (b) On appelle $\mathbf{X}^\star$, le minimum de $J$ sous la contrainte $\mathbf{X}^\star \in Q$.
   Démontrer que, nécessairement, $\nabla J(\mathbf{X}^\star) = 0$ où $\mathbf{X}^\star \in \partial Q$.
   (c) Mettre en œuvre la méthode du gradient projeté pour résoudre le problème de minimisation :

$$\min_{(x,y)\in Q} J(x,y).$$

   Pensez-vous que la méthode du gradient conjugué peut être associée à la méthode de projection ? Pourquoi ?
   (d) Représenter les itérés par cette méhode.
2. **S'il vous reste du temps.** Nous allons reprendre le même problème que précédemment, et évaluer une méthode de pénalisation. On propose les étapes suivantes :
   (a) Mettre en place une fonction de pénalisation $\mathbf{x} \mapsto \phi(\mathbf{x})$, en réfléchissant à l'expression qu'elle aura sur le cadrant $Q$. Déterminer alors le gradient de la fonctionnelle pénalisée.
   **Remarque :** attention au choix de la pénalisation ! La fonctionnelle pénalisée doit être différentiable.
   (b) Tracer les courbes de niveau de la nouvelle fonction coût et son gradient, pour plusieurs valeurs de $\varepsilon$. Que constatez-vous quant à la vitesse de variation de la fonction coût ? Dans la suite, on tracera le gradient uniquement sur le domaine admissible.
   (c) Pour une valeur petite de $\varepsilon$, par exemple $\varepsilon = 10^{-4}$ (pénalisation forte), et un point de départ $\mathbf{x}_0 = (-0.3,\ 0.5)^t$, tester la méthode de pénalisation pour les méthodes de gradient à pas fixe et à pas optimal. En visualisant les itérés, peut-on dire que la vitesse de convergence est satisfaisante ? Répéter le test pour $\varepsilon = 0.5$ (pénalisation faible). Que peut-on dire de la convergence ? Quid de la solution trouvée ?
   (d) Déduire de ces observations une méthode qui converge plus rapidement.

### 5.7.2 Travaux pratiques 2

*Dans cette séance de travaux pratiques, on souhaite utiliser les techniques classiques d'Optimisation avec contraintes pour résoudre deux problèmes appliqués : une équation aux dérivées partielles et la détermination de la distance d'un point à un hyperplan. On utilisera en particulier les algorithmes d'Uzawa et du gradient projeté.*

**Exercice 5.13** (Résolution d'un problème d'obstacle).

**Le modèle.**

Soit $g$, une fonction continue donnée sur le segment $[0,1]$. On considère un problème *d'obstacle* : trouver une fonction $u : [0,1] \longrightarrow \mathbb{R}$ telle que :

$$\begin{cases} -u''(x) \geq 1 & x \in (0,1) \\ u(x) \geq g(x) & x \in (0,1) \\ (-u''(x) - 1)\,(u(x) - g(x)) = 0 & x \in (0,1) \\ u(0) = u(1) = 0 \end{cases} \tag{5.6}$$

La première équation traduit une concavité minimale de la fonction $u$, la deuxième équation représente l'obstacle : on veut être au-dessus de $g(x)$. La troisième équation traduit le fait que l'on a au moins égalité dans une des deux équations précédentes : soit on résout $-u''(x) = 1$, soit $u(x) = g(x)$, et on est sur l'obstacle.

**Problème de minimisation associé.**

On discrétise ce problème en introduisant un maillage uniforme : $x_j = jh$, où $h$ désigne le pas en espace du maillage, et $j \in \{0, ..., n+1\}$, avec $n \geq 1$ entier et $h = \frac{1}{n+1}$. Posons pour $j \in \{0, ..., n+1\}$, $g_j = g(x_j)$. On cherche des valeurs $u_j = u(x_j)$, avec $j \in \{0, ..., n+1\}$, telles que :

$$\begin{cases} -\dfrac{u_{j-1} - 2u_j + u_{j+1}}{h^2} \geq 1 & j \in \{1, ..., n\} \\ u_j \geq g_j & j \in \{1, ..., n\} \\ \left(-\dfrac{u_{j-1} - 2u_j + u_{j+1}}{h^2} - 1\right)(u_j - g_j) = 0 & j \in \{1, ..., n\} \\ u_0 = u_{n+1} = 0 \end{cases} \tag{5.7}$$

On rappelle que $-\dfrac{u_{j-1} - 2u_j + u_{j+1}}{h^2}$ est l'approximation de $-u''(x_j)$ par la méthode des différences finies. Introduisons à présent la matrice $A \in \mathcal{M}_n(\mathbb{R})$, définie par :

$$A = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \vdots & \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}.$$

On appelle également $\mathbf{b}$ et $\mathbf{g}$, les vecteurs colonnes de $\mathbb{R}^n$ définis par :

$$\mathbf{b} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \text{ et } \mathbf{g} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}.$$

On introduit la notation $\mathbf{x} \geq \mathbf{y}$ pour des vecteur si $\forall i \in \{0, ..., n+1\}$, $x_i \geq y_i$.

Le résultat que je présente maintenant est une version discrétisée du théorème de Lax-Milgram que vous avez

probablement déjà rencontré dans le cadre de l'Analyse Fonctionnelle.

Soit

$$\mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}$$

On a l'équivalence :

$$\mathbf{u} \text{ est solution de } (5.7) \Longleftrightarrow \mathbf{u} \text{ est solution de } \begin{cases} \min\limits_{\mathbf{v}\in K} \left\{ \dfrac{1}{2}(A\mathbf{v},\mathbf{v}) - (\mathbf{b},\mathbf{v}) \right\} \\ K = \{\mathbf{v} \in \mathbb{R}^n : \mathbf{v} \geq \mathbf{g}\}. \end{cases}$$

On appelle alors $J$, la fonctionnelle définie sur $\mathbb{R}^n$ par :

$$J(\mathbf{v}) = \frac{1}{2}(A\mathbf{v},\mathbf{v}) - (\mathbf{b},\mathbf{v}).$$

**Résolution du problème.**
On se place ici dans le cas particulier où :

$$g(x) = \max\left(0, 1 - 100(x-0.7)^2\right).$$

On souhaite résoudre ce problème en utilisant l'algorithme du gradient projeté. On désigne par $\Pi_K$, la projection sur le convexe $K$. On pourra utiliser sans le démontrer que $\Pi_K(\mathbf{v}) = (\max(v_i, g_i))_{1\leq i\leq n}$.

1. (a) Écrire une méthode de gradient à pas fixe pour déterminer le minimum de $J$ sur $\mathbb{R}^n$. Tester cette méthode et vérifier qu'elle converge bien vers le résultat souhaité.

   (b) Adapter le programme précédent pour programmer la méthode du gradient projeté à pas constant. On testera notamment le programme pour $n = 10$. On se fixera un nombre maximal d'itérations et une tolérance de $10^{-5}$.

   (c) Demander à Matlab de calculer la première valeur propre de $A$, ainsi que la dernière, et comparer avec les résultats connus :

   $$\forall k \in \{1, ..., n\},\ \lambda_k(A) = \frac{4}{h^2}\sin^2\left(\frac{k\pi h}{2}\right).$$

   (d) Choisir $\rho > 0$ comme le pas optimal de la méthode de gradient à pas fixe sans contrainte.
   **Rappel :** $\rho_{\text{opt}} = \dfrac{2}{\lambda_1 + \lambda_n}$.

   (e) Afficher toutes les dix itérations : $k$ (le nombre itérations), l'estimateur d'erreur $\|\mathbf{u}^k - \mathbf{u}^{k-1}\|_2$ et la norme $\|J'(\mathbf{u}^k)\|$.

   (f) Afficher le graphe de $\mathbf{u}^k$ et de $\mathbf{g}$.

2. Faire tourner le programme pour $n = 100$, en adaptant le nombre maximal d'itérations.
   Noter le temps de calcul et le nombre d'itérations $k$ nécessaires pour obtenir $\|\mathbf{u}^k - \mathbf{u}^{k-1}\| \geq 10^{-6}$.
   On pourra également représenter le temps de calcul en fonction de $n$.
3. On désire calculer une approximation $\mathbf{u}^k$ de $\mathbf{u}$ avec une précsion de $10^{-4}$. On admet l'estimation suivante :

$$\|\mathbf{u}^k - \mathbf{u}\|_2 \leq \frac{\gamma}{1-\gamma}\|\mathbf{u}^k - \mathbf{u}^{k-1}\|_2, \text{ avec } \gamma := \|I - \rho A\|_2.$$

   Quelle tolérance $\eta$ doit-on choisir pour assurer que $\|\mathbf{u}^k - \mathbf{u}\|_2 \leq 10^{-4}$ ?
4. Conclure l'exercice.

**Exercice 5.14** (Distance d'un point à un plan).

Dans cet exercice, je vous laisse un peu plus libre qu'à l'ordinaire. On cherche à déterminer numériquement la plus courte distance entre un point $\mathbf{x}_0 \in \mathbb{R}^n$ et un hyperplan $(\mathcal{H})$ d'équation $A\mathbf{x} = \mathbf{b}$, où les lignes de la matrice $A$ sont linéairement indépendantes et $\mathbf{b}$, un vecteur colonne de taille $n$. Ce problème peut s'écrire comme un problème de programmation quadratique :

$$(\mathcal{P}) \quad \min_{A\mathbf{x}=\mathbf{b}} \frac{1}{2}.{}^t(\mathbf{x} - \mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0).$$

1. **Étude théorique.**
   (a) Montrer que le multiplicateur de Lagrange, à l'optimum est :

$$\boldsymbol{\lambda}^* = -(AA^t)^{-1}(b - A\mathbf{x}_0).$$

   (b) Montrer que la solution est :

$$\mathbf{x}^* = \mathbf{x}_0 + A^t(AA^t)^{-1}(\mathbf{b} - A\mathbf{x}_0).$$

   (c) Montrer que, dans le cas où $A$ est un vecteur ligne, la plus petite distance entre $\mathbf{x}_0$ et l'hyperplan vaut :

$$d(\mathbf{x}_0, \mathcal{H}) = \frac{||\mathbf{b} - A\mathbf{x}_0||}{\|A\|}.$$

2. **Étude numérique.**
   Faites-vous confirmer les résultats précédents à l'aide d'une étude numérique. On pourra par exemple utiliser l'algorithme d'Uzawa.

# Chapitre 6

# La méthode du recuit simulé

Nous nous plaçons dans le cas sans contrainte, mais la méthode du recuit simulé s'applique aussi dans le cas avec containtes.

Contrairement à toutes les méthodes vues jusqu'ici, le recuit simulé permet la recherche d'un minimum global. C'est son principal avantage. Ceci est possible car elle comporte une part d'aléatoire visant à explorer efficacement l'éventail des possibilités.
En contrepartie, la méthode est lente. C'est son principal inconvénient. Cet inconvénient est du reste commun à tous les algorithmes de recherche de minimum global (recuit simulé, algorithmes génétiques,..).

## 6.1 Principe

Il est inspiré de l'évolution d'un solide vers une position d'équilibre lors de son refroidissement.
Soit $S$ un système physique à la température $T$. On fait l'hypothèse que $S$ peut avoir un nombre dénombrable d'états $i \in \mathbb{N}$. A chaque état $i$, correspond un niveau d'énergie $E_i$. On note par $X$ l'état du système. On a alors :

**Theorem 6.1** *(loi de Boltzmann)*
*L'équilibre thermique est caractérisé par la distribution*

$$P_T(X = i) = \frac{1}{Z(T)} e^{-\frac{E_i}{k_B T}},$$

*où $k_B$ est la constante de Boltzmann et $Z$ est une fonction de normalisation donnée par : $Z(T) = \sum_{i \in \mathbb{N}} e^{-\frac{E_i}{k_B T}}$.*

Soit $i$ et $j$ deux états. Nous posons $\Delta E = E_i - E_j$. Il vient alors

$$\frac{P_T(X = i)}{P_T(X = j)} = e^{-\frac{\Delta E}{k_B T}},$$

et nous voyons que si $\Delta E < 0$, l'état $X = i$ est plus probable que $X = j$. Supposons à présent que $\Delta E > 0$, c'est bien sûr alors la situation inverse qui a lieu. Toutefois le rapport des probabilités dépend aussi de $k_B T$, et si $k_B T$ est grand devant $\Delta E$, les états $X = i$ et $X = j$ sont presque équiprobables.

Ainsi, à température élevée on passe facilement d'un état d'énergie à un autre (les molécules se meuvent facilement, le système présente un fort caractère aléatoire), mais quand on baisse progressivement la température, on fige la situation et on tend de plus en plus vers l'état d'équilibre du système (duquel on ne peut presque pas bouger).

Nous faisons à présent l'analogie suivante entre système physique et problème d'optimisation :
- états physiques $\leftrightarrow$ solutions admissibles
- énergie du système $\leftrightarrow$ côut d'une solution
- $k_B T \leftrightarrow$ paramètre de contrôle noté $T$.

C'est sur cette analogie que se base l'algorithme de Métropolis, en utilisant le principe physique évoqué précedemment.

## 6.2 L'algorithme de Métropolis

Soit $(S, f)$ un problème d'optimisation, et $i, j$ deux solutions admissibles. Nous introduisons le critère d'acceptation de $j$ par rapport à $i$ par :

$$P_T(\text{accepter } j) = \begin{cases} 1 & \text{si } \Delta f \geq 0 \\ e^{-\frac{\Delta f}{T}} & \text{sinon} \end{cases}$$

où $T \in \mathbb{R}^+$ est un paramètre de contrôle et $\Delta f = f(j) - f(i)$.

L'algorithme est défini à partir d'un état initial $i = i_0$ donné, et de la répétition de deux étapes (corps de l'algorithme). De plus, à fréquence régulière on fait baisser la température.
La première étape est appelée déplacement. Il s'agit de générer une solution admissible $j$ à partir de $i$. Dans une seconde étape, on met en jeu le critère d'acceptation pour savoir si $j$ est retenue ou non.

Nous donnons ci-dessous deux exemples de règle de déplacement :
- règle de Černy : On construit un repère $\{\mathbf{e_1}, .., \mathbf{e_n}\}$ en $j \in \mathbb{R}^n$, avec $\mathbf{e_1}$ pointant dans la direction du meilleur résultat obtenu. On choisit une direction $\mathbf{d}$ aléatoirement parmis $\mathbf{e_1}, .., \mathbf{e_n}$ avec une probabilité plus grande pour $\mathbf{e_1}$, et on se déplace aléatoirement suivant $\mathbf{d}$
- une autre règle : On tire aléatoirement un vecteur $\mathbf{u}$ dont chacune des composantes suit une loi uniforme

sur $[-1,1]$. On se déplace de $\delta j = q\mathbf{u}$, où $q$ est un pas fixé.

En ce qui concerne le refroidissement, on peut considérer par exemple l'une des règles suivantes :

– règle a : réduire $T$ à $(1-\epsilon)T$ tous les $m$ déplacements ($\epsilon$ et $m$ étant choisis par expérience), jusqu'à $T < T_{min} \approx 0$

– règle b : on se fixe un nombre total $K$ (très grand) de déplacements et on réduit $T$ après chaque ($k^{\text{eme}}$) $K/N$ déplacements en posant $T = T_k = T_0(1-k/N)^\alpha$. Ici $k \in \{1,2,..,N\}$, et on choisit en général $\alpha = 1, 2$ ou 4.

## 6.3 Travaux pratiques

**Exercice 6.1** (Recuit simulé).
On veut minimiser les fonctions

$$J_{1,\omega}(x) = 3 - x\sin(\omega\pi x)$$

sur l'intervalle $[-1,2]$, où $\omega = 2,4,...,10$, et, $J_2$ définie par

$$10\sin(0.3x\sin(1.3x^2 + 0.00001x^4 + 0.2x + 80))$$

sur $[-10,10]$. Tracer le graphe de ces fonction. Que donne la fonction Matlab **fminbnd** ?

On va programmer une méthode de recuit simulé pour essayer d'atteindre le minimum global. On propose

– de déplacer le point par une méthode de type Černy : pour chaque déplacement, on commence par choisir la direction.

1. Avec une probabilité de 2/3 on se dirige en direction du meilleur point obtenu jusque là (avec une proba de 1/3 dans l'autre direction). Bien sûr, si le point précédent est le meilleur point obtenu jusque là, il convient d'adapter l'algorithme.
2. Ensuite, on choisit la longueur du déplacement par une formule du type $b$.**rand** où $b$ est un paramètre qui pourra apparaitre comme l'une des variables de la fonction et **rand** un nombre tiré au hasard dans $[0,1]$ par Matlab.

– de baisser la température par la formule

$$T_k = T_0(1 - \frac{k}{N})^\alpha \qquad k = 1,2,\ldots,N$$

tous les $\frac{K}{N}$ mouvements et $\alpha$ pouvant prendre les valeurs 1, 2 ou 4. Les entiers $K$ et $N$ (ce dernier divisant $K$) seront aussi rentrés comme variables de la fonction.

Nous proposons le programme suivant. Tester ce code sur les exemples donnés et expérimenter.

```
% recuit simule avec deplacement de Cerny et un refroidissement fixe
% Entrees :
% J : fonction a minimiser sur l'intervalle [Xl ;Xr]
% T0 : temperature initiale
% alpha : ordre de la loi de refroidissement
% (N,K) : donnees du refroidissement
% b : longueur de deplacement
% Sortie
% x : minimum global approche
function x = recuit(J,Xl,Xr,T0,alpha,K,N,b) ;
% initialisation du point et de la temperature
x = (Xr-X1)*rand+X1 ; % x doit rester dans [Xl ;Xr]
Tk = T0 ;
bestpt = x ; % point qui minimise la fonction J
Jx = J(x) ; % evaluation
bestvalue = Jx ; % meilleur minimum
for k = 1 :N
     % formule du refroidissement
     Tk = T0*(1-k/N)^(alpha) ;
     display(x) ;
     for l = 1 :max(1,K/N)
         % Deplacement
         if (x == bestpt)
             % si x est le meilleur point : il faut en prendre
             % un autre aleatoirement qui se trouve a sa gauche mais
             % plus grand que Xl en partant de x comme origine
             x1 = max(Xl, x - b*rand) ;
         else % sinon on se deplace a sa gauche en partant de x de facon
             % aleatoire en partant de x et en pointant vers le meilleur
             % point, avec une longueur aleatoire rand * b, b fixe
             p1 = rand ;
             if (p1<2/3)
                 x1=max(Xl,min(Xr,x - sign(x-bestpt)*b*rand)) ;
             else
                 x1=max(Xl,min(Xr,x + sign(x-bestpt)*b*rand)) ;
             end
         end
         % on test ce nouveau point
         Jx1 = J(x1) ;
         DeltaJ = Jx1 - Jx ; % gradient d'energie
         % si x1 est meilleur que x, on le prend
         if (DeltaJ < 0)
             x = x1 ;Jx=Jx1 ;
         % et en plus on regarde si on a la meilleure valeur
             if (Jx1<bestval)
                 bestpt=x1 ;bestvalue=Jx1 ;
             end
```

```
    % si x1 est moins bon que x, on se permet sous condition de le choisir !
      elseif (rand < exp(-deltaJ/Tk))
           x=x1 ;Jx=Jx1 ;
       end
   end
end
```

---