Éléments d’optimisation

Pierre Bernhard

1er septembre 2001

Table des matières

1 Convexité **5**

1.0.1 Orientation . . . 5
1.0.2 Le cadre général . . . 5
1.1 Ensembles convexes . . . 6
1.1.1 Propriétés de base . . . 6
1.1.2 Projection sur un convexe . . . 10
1.2 Fonctions convexes . . . 16
1.2.1 Propriétés de base . . . 16
1.2.2 Régularité des fonctions convexes . . . 18
1.2.3 Sous-différentiel . . . 20
1.2.4 Dérivées et convexité . . . 22
1.3 Optimisation sous contraintes . . . 26
1.3.1 Le théorème de Kuhn et Tucker . . . 26
1.3.2 Le théorème de Lagrange . . . 30

2 Recherche unidimensionnelle **35**

2.1 Introduction . . . 35
2.1.1 Objectif . . . 35
2.1.2 Pente et dérivée numérique . . . 35
2.2 Méthodes directes . . . 36
2.2.1 Dichotomie . . . 36
2.2.2 Suites de Fibonacci . . . 37
2.2.3 Section dorée . . . 39
2.3 Méthodes indirectes . . . 40
2.3.1 “Backtracking” . . . 40
2.3.2 Méthode de Newton . . . 40
2.3.3 Approximation polynômiale . . . 41

3 Optimisation dans $\mathbb{R}^n$ **45**

3.1 Bonnes fonctions . . . 45
3.2 Optimisation non contrainte . . . 45
3.2.1 Relaxation . . . 45
3.2.2 Gradient à pas optimal . . . 47
3.2.3 Méthode de Newton . . . 49
3.3 Optimisation sous contraintes inégalité . . . 51
3.3.1 Position du problème . . . 51

3.3.2 Gradient projeté 52
3.3.3 Algorithme d'Uzawa 54
3.3.4 Pénalisation 56
3.3.5 Méthode du chemin central 58
3.4 Optimisation sous contraintes égalité 60
3.4.1 Contraintes affines 60
3.4.2 Contraintes nonlinéaires 62

4 Programmation linéaire et programmation dynamique 65
4.1 Programmation linéaire 65
4.1.1 Position du problème 65
4.1.2 Étude du polyèdre 67
4.1.3 L'algorithme du simplexe 69
4.1.4 Rudiments de dualité 71
4.2 Programmation dynamique 72
4.2.1 Plus court chemin dans un graphe orienté 73
4.2.2 Système dynamique et programmation dynamique 75

Chapitre 1

Convexité

1.0.1 Orientation

Ce chapitre introductif à un cours d'optimisation, lui même axé sur les algorithmes, introduit quelques notions de convexité qui seront utiles pour la suite. Ce n'est en aucun cas un "traité" de convexité. En particulier, nous ignorons —hélas— tout ce qui a trait à la transformée de Fenchel, et donc à la dualité, et au théorème de Von Neumann-Sion.

Nous présentons l'optimisation dans son cadre naturel, c'est à dire non différentiable. Mais par souci de ne présenter que les méthodes de base en optimisation, nous nous limiterons pratiquement, dans les chapitres suivants, à l'optimisation de fonctions dérivables. Aussi, ce chapitre fera-t-il le lien entre les deux conceptions, réconciliant la vieille notion qu'une fonction convexe est une fonction "dont la dérivée seconde est positive" avec une présentation essentiellement "non différentiable" de la convexité.

On terminera par les applications fondamentales de la convexité au plan théorique en optimisation, culminant avec le théorème de Khun et Tucker. Enfin, par souci d'être raisonnablement complet et de ne pas séparer ce dernier théorème de son contexte naturel, nous terminons par le théorème des multiplicateurs de Lagrange, dont nous donnons la preuve la plus classique, via le théorème des fonctions implicites.

La seule ambition de tout cela est d'introduire les chapitres suivants, sur les *algorithmes*.

1.0.2 Le cadre général

La variable notée x, y, etc., évoluera dans un ensemble X dont on évitera par la suite de préciser s'il est de dimension finie ou infinie. C'est dire que le lecteur peut toujours choisir de lire ce texte en considérant que X est $\mathbb{R}^n$, et alors si $p \in \mathbb{R}^n$ également, on aura naturellement

$$(p,x) = \sum_{i=1}^{n} p_i x_i \,, \qquad \text{et} \qquad \|x\|^2 = (x,x)$$

désignera (le carré de) la norme euclidienne classique.

Par contre, le lecteur plus ambitieux peut choisir de voir en X un espace de Hilbert de dimension infinie. Nous soulignerons les rares endroits où une démonstration doit être modifiée pour ce cas, voire les deux théorèmes qui ne seront vrais qu'en dimension finie.

Nous choisissons d'appeler u une fonction réelle de X dans $\mathbb{R}$, dont on décrira les propriétés de convexité, par homogénéité avec la suite où la fonction à minimiser s'appellera u, réservant f pour les contraintes (qui seront elles aussi convexes en général).

1.1 Ensembles convexes

1.1.1 Propriétés de base

Premières définitions

La convexité est une théorie *géométrique*, dont l'objet géométrique de base est le *segment* :

Définition 1.1 (Segment) *Soient x et y deux points de X, on appelle* segment $[x, y]$ *l'ensemble des points de la forme*

$$[x, y] = \{(1 - \lambda)x + \lambda y \mid \lambda \in [0, 1]\}. \tag{1.1}$$

(Ici, $[0, 1]$ désigne l'intervalle fermé de $\mathbb{R}$), et on dit que ce segment *relie x et y*.

Bien-sûr, si on est dans $\mathbb{R}^2$ ou $\mathbb{R}^3$, il s'agit bien ici du segment au sens élémentaire de la géométrie euclidienne. Il faut toujours voir la convexité comme une extension à X (soit $\mathbb{R}^n$ soit un Hilbert) de concepts géométriques de $\mathbb{R}^2$ ou $\mathbb{R}^3$.

Une remarque importante (bien que banale) est la suivante :

Remarque 1.1 *Le segment défini par (1.1) est identiquement décrit par*

$$[x, y] = \{\lambda x + (1 - \lambda)y \mid \lambda \in [0, 1]\}.$$

Il suffit en effet de remarquer que λ de cette dernière formulation est juste $(1 - \lambda)$ de la première, qui parcourt le même intervalle $[0, 1]$.

Nous introduisons maintenant la convexité :

Définition 1.2 (Ensemble convexe) *Un sous-ensemble C de X est dit* convexe *si chaque fois qu'il contient deux points il contient le segment qui les relie.*

Cette définition se lit également

Définition 1.3 (Ensemble convexe) *Un sous-ensemble C de X est dit* convexe *si*

$$\{x \in C, y \in C\} \Rightarrow [x, y] \subset C$$

ou de manière équivalente,

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

On dira souvent "un convexe" pour "un sous-ensemble convexe".

On déduit immédiatement de ces définitions le fait suivant :

Proposition 1.1 *L'intersection d'ensembles convexes est convexe.*

La démonstration est élémentaire. On invite seulement le lecteur à vérifier qu'elle ne se limite pas à une intersection d'un nombre fini d'ensembles.

De cette dernière remarque découle la possibilité de donner une autre définition. Soit A un sous ensemble quelconque de X.

Définition 1.4 (Enveloppe convexe) *On appelle* enveloppe convexe *de A, notée co(A), l'intersection de tous les sous-ensembles convexes contenant A.*

D'après la propriété qui précède, co(A) est un ensemble convexe, et clairement il contient A. C'est *le plus petit convexe contenant A*. En effet, il est, par définition, contenu dans tout autre convexe contenant A.

On devrait logiquement introduire ici la définition d'un *point extrémal* d'un convexe. Mais nous ne nous en servirons que pour la programmation linéaire, et nous n'introduirons ce concept qu'à ce moment là.

Combinaisons convexes

Nous introduisons maintenant un concept qui étend en quelque sorte celui de segment.

Définition 1.5 (Combinaison convexe) *On appelle* combinaison convexe *de p points $\{x_1, x_2, \dots x_p\}$ de X un point de X de la forme*

$$x = \sum_{i=1}^{p} \lambda_i x_i \,, \qquad \lambda_i \geq 0 \quad \forall i \,, \qquad \sum_{i=1}^{p} \lambda_i = 1 \,.$$

Une combinaison convexe de p points est donc caractérisée, outre ces p points, par *un vecteur du simplexe de $\mathbb{R}^p$*, c'est à dire par p nombres positifs ou nuls (nécessairement inférieurs ou égaux à 1) sommant à 1. On remarque en particulier que le segment $[x, y]$ est l'ensemble des combinaisons convexes de x et y.

Insistons sur le fait qu'une combinaison convexe n'est définie ici que pour un nombre *fini* de points. (Quand nous dirons "combinaisons convexes", le lecteur peut toujours ajouter "finies".)

Remarquons alors qu'on a une définition alternative d'un convexe :

Proposition 1.2 (Propriété caractéristique) *Un ensemble est convexe si et seulement si il contient toutes les combinaisons convexes de ses points.*

Preuve C'est manifestement suffisant, car alors il contient les combinaisons convexes de ses paires de points, les segments.

Quant au caractère nécessaire, remarquons le tout petit lemme suivant :

Lemme 1.3 *Une combinaison convexe de p points peut être représentée comme un élément du segment joignant le pième point à une combinaison convexe des $p-1$ premiers.*

Preuve du lemme Soit

$$x = \sum_{i=1}^{p} \lambda_i x_i$$

une combinaison convexe. Il suffit de remarquer que $\sum_{i=1}^{p-1} \lambda_i = 1 - \lambda_p$ et de poser

$$\mu_i = \frac{\lambda_i}{1 - \lambda_p} \,, \quad x' = \sum_{i=1}^{p-1} \mu_i x_i \,.$$

On laisse le lecteur vérifier que x' est bien une combinaison convexe des $p-1$ premiers x_i, et finalement que $x = (1 - \lambda_p)x' + \lambda_p x_p$, ce qui prouve le lemme.

Donc, comme le convexe doit contenir les combinaisons convexes de deux de ses points, en prenant l'une d'elle et sa combinaison convexe avec un troisieme, qui doit appartenir au convexe comme combinaison convexe de deux de ses points, on a toutes les combinaisons convexes de ces trois points, et ainsi de suite.

Le lemme ci-dessus est précisé par la propriété suivante :

Proposition 1.4 *Une combinaison convexe de combinaisons convexes est une combinaison convexe.*

On laisse le lecteur faire cette vérification particulièrement fastidieuse. Elle est facile une fois qu'on a remarqué que

– On peut considérér que toutes les combinaisons convexes dont on fait la combinaison convexe s'appuient sur le même ensemble de points (et ont donc la même "longueur"). Il suffit de prendre l'union de tous les points concernés et de compléter les combinaisons convexes par des coefficients nuls en tant que de besoin,
– On n'a pas besoin, dans la définition d'une combinaison convexe, que les x_i soient distincts.

Une conséquence importante de cette dernière proposition est la suivante :

Théorème 1.5 *L'enveloppe convexe d'un ensemble A peut être représentée comme l'union de toutes les combinaisons convexes de ses points.*

Démonstration Comme convexe contenant A, co(A) doit nécessairement contenir l'ensemble décrit dans le théorème. Grâce à la proposition précédente, cet ensemble est convexe lui-même, et il contient manifestement A. Donc il contient co(A). Il est donc égal à co(A).

Deux propriétés utiles

On signale ici deux résultats utiles sans être essentiels pour notre propos. Le premier traite des proprétés topologiques des convexes, or nous aurons souvent à considérer l'intérieur d'un convexe.

Lemme 1.6 *Soit C un sous-ensemble convexe. Si $x \in \overset{\circ}{C}$ (l'intérieur de C) et $y \in C$, alors le segment $[x, y]$ privé de y (que nous noterons $[x, y)$) est contenu dans $\overset{\circ}{C}$.*

Démonstration La démonstration consiste à placer une petite boule contenue dans C autour de x, et par homothétie de centre y en déduire l'existence d'une boule contenue dans C centrée en tout point de $[x, y)$.[1]

Soit donc ε un nombre positif tel que la boule de centre x et de rayon ε soit contenue dans C. Soit aussi $\lambda \in (0, 1]$ et $x' = \lambda x + (1 - \lambda)y$. Soit enfin $z' \in X$ dans la boule de centre x' et de rayon $\lambda\varepsilon$. Nous allons montrer que z' est dans C, ce qui implique que toute la boule de centre x' et de rayon $\lambda\varepsilon$ est dans C, établissant le lemme.

Construisons $z = y + (1/\lambda)(z' - y)$. Remarquons que $x = y + (1/\lambda)(x' - y)$, de sorte que $z - x = (1/\lambda)(z' - x')$. Comme par hypothèse, $\|z' - x'\| \leq \lambda\varepsilon$, il en découle que $\|z - x\| \leq \varepsilon$ et donc que $z \in C$. Remarquons enfin que $z' = \lambda z + (1 - \lambda)y$ est une combinaison convexe de z et y qui sont tous les deux dans C. Donc $z' \in C$ et le lemme est démontré.

Nous voulons insister sur la nature de cette démonstration. On fait une démonstration géométrique dans $\mathbb{R}^2$, évoquée en exergue de la démonstration ci-dessus, et on la traduit en un calcul qui en fait une démonstration dans X seulement doté de sa structure d'espace de Hilbert (ou Euclidienne). Cette démarche sera au cœur de démonstrations de propriétés plus difficiles.

L'intérêt de ce lemme est dans ce corollaire :

Corollaire 1.7 *Soit C un convexe d'intérieur non vide, alors son adhérence est l'adhérence de son intérieur : $\bar{C} = \overline{\overset{\circ}{C}}$.*

Preuve Soit $y \in \bar{C}$, prenons $x \in \overset{\circ}{C}$ et remarquons que $y = \lim_{\lambda \to 0}(\lambda x + (1 - \lambda)y)$, tous points de $\overset{\circ}{C}$ aussi longtemps que $\lambda \neq 0$ d'après le lemme.

Nous abordons enfin un des rares théorèmes de cette théorie qui soit spécifiquement en dimension finie. Il généralise à $\mathbb{R}^n$ la remarque que si x est dans un polygône plan, il est dans un des triangles que les sommets de ce polygône définissent. Il ne nous servira pas vraiment dans la suite, mais c'est un classique...

[1] Il devrait être *interdit* d'écrire un texte sur la convexité sans figure. L'auteur espère réparer un jour sa transgression.

Théorème 1.8 (Carathéodory) *Une combinaison convexe de p points de $\mathbb{R}^n$, $p > n+1$, est combinaison convexe de $n+1$ d'entre-eux.*

Démonstration Soient $x_0, x_1, \ldots, x_p$ $p+1$ points de $\mathbb{R}^n$ avec $p > n$, et

$$x = \sum_{i=0}^{p} \lambda_i x_i \tag{1.2}$$

une combinaison convexe. Il nous faut exhiber $n+1$ points x_{i_k} et autant de coefficients μ_k d'une combinaison convexe tels que

$$x = \sum_{k=0}^{n} \mu_k x_{i_k} .$$

Puisque nous sommes dans $\mathbb{R}^n$ et que $p \geq n+1$, les vecteurs $x_i - x_0$, $i = 1, \ldots, p$ sont linéairement dépendants. Il existe donc un jeu de coefficients α_i non tous nuls tels que

$$\sum_{i=1}^{p} \alpha_i (x_i - x_0) = 0,$$

équation qui demeure si on multiplie tous les α_i par le même nombre (non nul) ν.

Nous écrivons (1.2) sous la forme

$$x - x_0 = \sum_{i=1}^{p} \lambda_i (x_i - x_0) ,$$

et nous ajoutons au membre de droite la quantité nulle exhibée avant :

$$x - x_0 = \sum_{i=1}^{p} (\lambda_i + \nu\alpha_i)(x_i - x_0) . \tag{1.3}$$

Il retse à montrer qu'on peut ajuster ν pour que les $\lambda_i + \nu\alpha_i$ soient tous positifs sauf un qui est nul, et somment à moins que 1. Ainsi ce seront nos μ_i, $i = 1, \ldots, p$ et μ_0 sera défini par différence de la somme à 1.

Ceci est accompli de la façon suivante. Au prix de changer s'il le faut tous les α_i en $-\alpha_i$, assurons nous que

$$\sum_{i=1}^{p} \alpha_i \geq 0 .$$

Soit j l'indice tel que $\alpha_j > 0$, et λ_j/α_j est minimum parmi tous les rapports λ_k/α_k positifs (i.e. avec $\alpha_k > 0$). Prenons alors $\nu = -\lambda_j/\alpha_j$, et posons

$$\mu_i = \lambda_i + \nu\alpha_i .$$

On remarque d'abord que les μ_i sont tous positifs ou nuls. En effet, soit $\alpha_i \leq 0$, et alors c'est évident (ν est négatif), soit $\alpha_i > 0$, mais alors

$$\mu_i = \lambda_i + \nu\alpha_i = (\frac{\lambda_i}{\alpha_i} - \frac{\lambda_j}{\alpha_j})\alpha_i$$

et la parenthèse est positive ou nulle par le choix de j, et donc aussi μ_i.

On remarque ensuite que puisque les α_i ont une somme positive ou nulle, et ν est négatif, les μ_i, $i = 1, \ldots, p$ ont une somme inférieure aux λ_i pour les mêmes indices, donc inférieure à 1. (Et même à $1 - \lambda_0$.)

Ainsi les μ_i avec $\mu_0 = 1 - \sum_1^p \mu_i$ forment les coefficients d'une combinaison convexe et (1.3) représente x comme une combinaison convexe des x_i avec les μ_i pour coefficients. Mais maintenant, $\mu_j = 0$, et il n'y a donc plus que p (et non plus $p + 1$) termes dans cette combinaison. Si $p > n + 1$ on recommence.

Ce théorème classique a des conséquences sur la géométrie des polyèdres, mais nous ne développerons pas cette direction. Il montre aussi que l'enveloppe convexe d'un ensemble de $\mathbb{R}^n$ peut être obtenue comme union de toutes les combinaisons convexes de $n + 1$ (ou moins) de ses points.

1.1.2 Projection sur un convexe

Pour simples que soient les quelques résulats de ce paragraphe, ils sont au centre de ce qui fait la puissance du concept de convexité. Un concept géométrique qui nous sera très utile est celui d'angle aigu ou d'angle obtu. Nous l'introduisons formellement ici pour insister sur son utilité :

Définition 1.6 (Angle obtu) *Deux vecteurs (autrement appelés "points", mais la référence à un vecteur est plus intuitive ici) u et v de X seront dits former un angle obtu si leur produit scalaire est négatif ou nul, être orthogonaux si leur produit scalaire est nul, et former un angle aigu si leur produit scalaire est positif ou nul.*

(On remarque donc que, par commodité, on a admis les angles droits parmi les angles obtus et les angles aigus.)

Le théorème de projection

Nous rappelons la définition de la distance d'un point à un sous-ensemble :

Définition 1.7 (Distance à un sous-ensemble) *On appelle distance d'un point x de X à un sous-ensemble A, et on note $d(x, A)$,*

$$d(x, A) = \inf_{y \in A} \|y - x\| .$$

On fait remarquer en outre que $x \in \bar{A}$ (et donc si A est fermé $x \in A$) si et seulement si $d(x, A) = 0$.

Théorème 1.9 (Projection sur un convexe fermé) *Soit C un sous-ensemble convexe fermé de X. Soit $x \in X$. Il existe un et un seul point $\hat{x}$ de C tel que $d(x, C) = \|x - \hat{x}\|$. Ce point est appelé* projection de x sur C *et noté $P_C(x)$.*

Démonstration Faisons d'abord remarquer que si $x \in C$, le résultat est banal, avec $P_C(x) = x$. Nous nous intéressons donc au cas où $x \notin C$. La démonstration dépend du lemme géométrique suivant :

Lemme 1.10 *Soit C convexe fermé, $x \notin C$, $y_1, y_2 \in C$, l'un des y_i, pour $i = 1$ ou 2 satisfait*

$$\|x - y_i\|^2 \geq d(x, C)^2 + \frac{1}{4}\|y_1 - y_2\|^2 \tag{1.4}$$

Preuve du lemme Introduisons le milieu $y_0 = (y_1 + y_2)/2$ du segment $[y_1, y_2]$. Alors, $y_2 - y_0 = -(y_1 - y_0)$. Donc l'un au moins de ces deux vecteurs forme un angle obtu avec $x - y_0$, soit $(x - y_0, y_i - y_0) \leq 0$. (Puisque ces produits scalaires sont opposés pour $i = 1$ et $i = 2$.) On a ainsi

$$\|x - y_i\|^2 = \|(x - y_0) - (y_i - y_0)\|^2 = \|x - y_0\|^2 - 2(x - y_0, y_i - y_0) + \|y_i - y_0\|^2 .$$

Mais, par la convexité de C, $y_0 \in C$. Donc $\|x - y_0\|^2 \geq d(x, C)^2$. On a choisi i pour que $(x - y_0, y_i - y_0) \leq 0$, et manifestement $\|y_i - y_0\|^2 = (1/4)\|y_1 - y_2\|^2$. Le lemme en découle.

Revenons à la preuve du théorème. Soit $\{y_k\}$ une suite de points de C telle que $\|x - y_k\| \to d(x, C)$. Grâce au lemme, on peut affirmer que cette suite est de Cauchy. En effet, soit $\varepsilon > 0$, on peut choisir N suffisamment grand pour que, pour tout n et m supérieur à N, $\|x - y_n\|^2$ et $\|x - y_m\|^2$ soient inférieurs à $d(x, C)^2 + \varepsilon^2/4$. En appliquant le lemme, il en découle que $\|y_n - y_m\| \leq \varepsilon$.

Donc la suite converge vers un point $\hat{x}$. Comme C est fermé, $\hat{x} \in C$. Comme la norme est continue, $\|x - \hat{x}\| = d(x, C)$. Le lemme appliqué à deux points qui satisferaient cette propriété implique qu'ils sont confondus. Le théorème est démontré.

Remarque 1.2 *On caractérise $P_C(x)$ comme le point de C le plus proche de x.*

Une importante caractérisation de la projection est donnée par le résultat suivant

Théorème 1.11 (Angle obtu) *Soit C un convexe fermé de X et $x \notin C$. Alors $\hat{x} = P_C(x)$ si et seulement si $\hat{x} \in C$ et*

$$\forall y \in C , \quad (x - \hat{x}, y - \hat{x}) \leq 0 . \tag{1.5}$$

Démonstration La condition est suffisante. Supposons en effet (1.5) vérifié. Prenons $x' \in C$ différent de $\hat{x}$. Comme dans le lemme, on écrit $\|x - x'\|^2 = \|(x - \hat{x}) - (x' - \hat{x})\|^2$ et on développe le carré. Le fait que $(x - \hat{x}, x' - \hat{x}) \leq 0$ implique que $\|x - x'\|^2 \geq \|x - \hat{x}\|^2 + \|x' - \hat{x}\|^2$, établissant bien $\hat{x}$ comme le point de C le plus proche de x.

Réciproquement, supposons que pour un certain $\hat{x} \in C$, il existe $y \in C$ tel que $(x - \hat{x}, y - \hat{x}) \geq 0$. Considérons les points x_t de la forme $x_t = \hat{x} + t(y - \hat{x})$. Par la convexité de C, pour $t \in [0, 1]$, $x_t \in C$. Mais par le même calcul que précédemment on a

$$\|x - x_t\|^2 = \|x - \hat{x}\|^2 - 2t(x - \hat{x}, y - \hat{x}) + t^2\|y - \hat{x}\|^2 .$$

Pour t suffisamment petit positif, le terme en t domine celui en t^2 et impose son signe à la différence, de sorte qu'alors $\|x - x_t\|^2 < \|x - \hat{x}\|^2$. Donc $\hat{x}$ ne saurait être la projection de x sur C. Ce qui achève de démontrer le théorème.

Remarquons que si $x \notin C$, nécessairement $P_C(x) \in \partial C$. (exercice). Ceci mène à l'important concept suivant :

Définition 1.8 (Normale extérieure) *Soit C un sous-ensemble convexe et x un point de ∂C. On appelle* normale extérieure à C en x *tout vecteur ν de X tel que*

$$\forall y \in C , \quad (\nu, y - x) \leq 0 . \tag{1.6}$$

Au vu de cette définition, le théorème de l'angle obtu, ou l'inégalité 1.5, se dit aussi : si $x \notin \bar{C}$, $x - P_C(x)$ est une normale extérieure à C. D'où on peut déduire que par tout x extérieur à un convexe passe une normale extérieure à ce convexe. La réciproque, à savoir qu'en tout point de ∂C il existe au moins une normale extérieure, sera une conséquence du deuxième théorème de séparation ci-dessous.

Une propriété importante de la projection sur, ou des normales extérieures à, un convexe est la propriété de contraction suivante :

Théorème 1.12 ("Pont suspendu") *Soient C un convexe fermé, x_1 et x_2 deux points de $\mathbb{R}^n$, et $\hat{x}_1$ et $\hat{x}_2$ leurs projections respectives sur C. On a*

$$\|\hat{x}_2 - \hat{x}_1\| \leq \|x_2 - x_1\| . \tag{1.7}$$

Démonstration On applique la propriété de l'angle obtu aux deux projections, avec chaque fois l'autre projeté pour point "y" de C :

$$\begin{aligned} (\hat{x}_2 - \hat{x}_1, x_1 - \hat{x}_1) &\leq 0 , \\ (\hat{x}_1 - \hat{x}_2, x_2 - \hat{x}_2) &\leq 0 . \end{aligned}$$

En changeant le signe des deux membres du deuxième produit scalaire ci-dessus, et en ajoutant, il vient

$$(\hat{x}_2 - \hat{x}_1, x_1 - x_2 + \hat{x}_2 - \hat{x}_1) \leq 0 .$$

(On a réordonné le terme de droite du produit scalaire.) On écrit cela

$$(\hat{x}_2 - \hat{x}_1, \hat{x}_2 - \hat{x}_1) \leq (\hat{x}_2 - \hat{x}_1, x_2 - x_1) ,$$

soit en utilisant Cauchy-Schwarz pour majorer le terme de droite

$$\|\hat{x}_2 - \hat{x}_1\|^2 \leq \|\hat{x}_2 - \hat{x}_1\| \|x_2 - x_1\| .$$

Soit $\|\hat{x}_2 - \hat{x}_1\| = 0$, et alors l'inégalité (1.7) est vérifiée, soit on peut diviser de part et d'autre par $\|\hat{x}_2 - \hat{x}_1\|$, et il reste précisément l'inégalité (1.7).

Théorèmes de séparation

Grâce à la notion de projection, nous obtenons facilement les théorèmes de séparation ci-dessous. Ces théorèmes, célèbres, sont encore vrais dans un cadre non hilbertien (espaces de Banach, c'est à dire sans produit scalaire), mais alors ils sont difficiles, et équivalents au célèbre théorème de Hahn Banach.

Nous avons besoin d'un autre concept géométrique, celui d'hyperplan et de demi-espaces délimités par un hyperplan. Ceci étend aux hilberts la remarque simple qu'une droite de $\mathbb{R}^2$ ou un plan de $\mathbb{R}^3$ séparent l'espace où ils vivent en deux demi-espaces disjoints.

Définition 1.9 (Hyperplan, demi-espace) *On appelle* hyperplan *défini par un vecteur $p \in X$ et un réel a l'ensemble des points $\{x \in X \mid (p, x) = a$. On appelle demi-espaces fermés les deux sous ensembles $\{x \in X \mid (p, x) \geq a\}$ et $\{x \in X \mid (p, x) \leq a\}$.*

Les demi-espaces ouverts sont définis de la même façon, mais avec des inégalités strictes.

On laisse le lecteur vérifier la proposition suivante :

Proposition 1.13 *On a les propriétés suivantes, où H désigne l'hyperplan $(p, x) = a$:*

- *les hyperplans sont des convexes fermés,*
- *en tout point de H, p et $-p$ sont les deux seules normales extérieures à H,*
- *$\forall x \in X, \forall y, z \in H, (x - P_H(x), y - z) = 0$,*
- *la projection sur un hyperplan est linéaire.*

Théorème 1.14 (Séparation stricte) *Étant donnés un sous-ensemble convexe C de X et un point $x \notin C$, il existe un hyperplan qui* sépare *strictement x et C, c'est à dire tel que x et C soient contenus dans les deux différents demi-espaces ouverts définis par cet Hyperplan.*

Démonstration Il suffit de prendre un hyperplan orthogonal à $x - P_C(x)$ passant par le point milieu, soit, avec les notations de la définition ci-dessus, $p = x - P_C(x)$, $a = (1/2)(\|x\|^2 - \|P_C(x)\|^2)$. On laisse le lecteur vérifier, en se servant de l'inégalité $\|x\|^2 + \|\hat{x}\|^2 \geq 2(x, \hat{x})$ pour tout $x \neq \hat{x}$.

On laisse en exercice démontrer qu'un sous-ensemble compact convexe et un convexe fermé disjoints peuvent être séparés strictement par un hyperplan.

Un théorème plus difficile est le suivant :

Théorème 1.15 (Séparation large) *Soit C un sous-ensemble convexe (non vide)* ouvert, *soit $x \notin C$. Il existe un hyperplan qui contient x et laisse C dans un demi-espace ouvert.*

Remarque 1.3 *Remarquons que cette affirmation n'est intéressante que si x est sur la frontière de C. En effet, si-non il suffit de projeter x sur $\bar{C}$, et d'utiliser $x - P_C(x)$ pour normale à l'hyperplan. La normale à l'hyperplan sera une normale extérieure à C (exercice). Donc ce théorème est équivalent à la proposition suivante :*

Proposition 1.16 *En tout point $\bar{x}$ de la frontière d'un convexe d'intérieur non vide C, il y a au moins une normale extérieure, c'est à dire un vecteur $\nu \in X$ tel que*

$$\forall y \in C \quad (\nu, y - x) \leq 0\,.$$

Démonstration La démonstration présente une difficulté particulière dans le cas où X est de dimension infinie. Nous la donnons dans le cas où $X = \mathbb{R}^n$, puis nous indiquerons ensuite où se situe la difficulté et comment la contourner si X est un Hilbert quelconque.

Soit donc C un sous-ensemble convexe ouvert non vide, et $x \in \partial C$. Comme C est ouvert, on peut prendre un z à l'intérieur de C. En particulier, il existe un $\varepsilon > 0$ tel que la boule (fermée) de centre z et de rayon ε soit contenue dans C. Soit

$$x_1 = 2x - z\,, \qquad \text{soit} \quad x = \frac{1}{2}x_1 + \frac{1}{2}z\,.$$

Alors, $x_1 \notin C$. Car si x_1 était dans C, comme z y est et que C est convexe, x y serait, ce qui contredit l'hypothèse.

Considérons les points $x_t = x + t(x_1 - x)$, pour $t > 0$ (destiné à tendre vers zéro). On a

Lemme 1.17 *La boule de centre x_t et de rayon $t\varepsilon$ n'intersecte pas C.*

Preuve du lemme En effet, supposons au contraire que $y \in C$ avec $\|y - x_t\| \leq t\varepsilon$. Considérons le point homothétique de y dans l'homothétie de centre x et de rapport $-1/t$. Soit $y' = x - (1/t)(y - x)$. En remarquant que $z = x - (x_1 - x)$, il vient $y' - z = x_1 - x - (1/t)(y - x)$ tandis que $y - x_t = y - x - t(x_1 - x)$. De sorte que comme l'homothétie devait l'entrainer, $y' - z = -(1/t)(y - x_t)$. Donc $\|y' - z\| \leq \varepsilon$, et $y' \in C$. Enfin, comme $x = [1/(1+t)]y + [t/(1+t)]y'$ est une combinaison convexe de y et y' (y' a été construit pour ça) et que dans notre hypothèse y et y' sont tous les deux dans C, x devrait y être aussi, contredisant l'hypothèse de départ. Donc $y \notin C$, et le lemme est démontré.

Ceci établit que pour tout $t > 0$, les x_t ne sont pas dans $\bar{C}$, puisqu'ils sont centres d'une boule de rayon positif qui n'intersecte pas C.

Ainsi, on peut considérer leur projection $\hat{x}_t$ sur $\bar{C}$ et le vecteur normalisé correspondant $p_t = (x_t - \hat{x}_t)/\|x_t - \hat{x}_t\|$. Quand $t \to 0$, $x_t \to x$, et par le théorème 1.12, $\hat{x}_t \to x$ (car x est sa propre projection sur $\bar{C}$). Les p_t parcourent la sphère unité, qui est compacte. (C'est ici que la preuve en dimension infinie diverge.) Ils ont donc au moins un point d'accumulation p non nul (de norme 1).

Soit t_k une suite tendant vers 0 quand $k \to \infty$, et telle que $p_{t_k} \to p$. Prenons un y fixe dans C. On a, par le théorème de l'angle obtu,

$$0 \geq (y - \hat{x}_{t_k}, p_{t_k}) \to (y - x, p) \tag{1.8}$$

(par continuité du produit scalaire par rapport à ses arguments, continuité que l'inégalité de Cauchy-Schwarz établit) ce qui établit que $(y - x, p) \leq 0$, ou $(y, p) \leq (x, p)$.

Il reste à obtenir l'inégalité stricte. Comme C est ouvert, en tout y de C on peut centrer une boule de rayon α, dépendant de y mais strictement positif, contenue dans C. Ainsi, $y + \alpha p \in C$ (rappelons que $\|p\| = 1$). En appliquant l'inégalité large à ce point de C on obtient l'inégalité stricte sur y, comme désiré.

Complément Pour les espaces de Hilbert. La boule unité est seulement *faiblement* compacte, donc on peut seulement assurer l'existence d'une suite t_k telle que les p_{t_k} tendent faiblement vers un p. La limite (1.8) reste correcte, la convergence de $\hat{x}_{t_k}$ vers x étant forte. Mais il reste à démontrer que p n'est pas nul. Ceci est la conséquence du lemme suivant, qui garde son intérêt en dimension finie :

Lemme 1.18 *On a pour tout $t > 0$*

$$(p_t, x - z) \geq \varepsilon\,.$$

Par passage à la limite (faible), on en déduira que $(p, x - z) \geq \varepsilon$ et donc que $p \neq 0$.

Preuve du lemme. On appelle $\bar{x}_t$ la projection de x sur la droite support de p_t passant par $\hat{x}_t$ (et x_t donc) : $\bar{x}_t = x_t - (p_t, x_t - x)p_t$. La propriété de l'angle obtu donne $(p_t, x) \leq (p_t, \hat{x}_t)$. On a ainsi (la première inégalité ci-dessous est Cauchy-Schwarz)

$$\|x_t - \bar{x}_t\| \geq (p_t, x_t - \bar{x}_t) = (p_t, x_t - x) \geq (p_t, x_t - \hat{x}_t) = \|x_t - \hat{x}_t\|\,.$$

D'après le lemme précédent, la dernière distance ci-dessus n'est pas moindre que $t\varepsilon$ puisque $\hat{x}_t \in \bar{C}$. Or $\|x_t - \bar{x}_t\| = (p_t, x_t - x)$ par construction. Le lemme est démontré en remarquant que $x_t - x = t(x - z)$.

Complément mathématique

Pour les amateurs de mathématiques, voici quelques compléments culturellement essentiels, mais pas indispensables pour un cours d'optimisation dans $\mathbb{R}^n$. Ce sont deux conséquences du théorème de projection.

Le premier résultat concerne les *formes linéaires* de X. On appelle "forme linéaire" une application linéaire continue de X dans $\mathbb{R}$. Dans le cas de dimension finie, disons n, on sait bien qu'une forme linéaire est définie par sa "matrice", une ligne de n coefficients. On peut donc la voir comme un vecteur de $\mathbb{R}^n$. En dimension infinie, à tout vecteur, disons y de X on peut encore faire correspondre une forme linéaire par $x \mapsto a(x) = (y, x)$. Ce que dit le théorème qui suit est qu'il n'y a pas d'autres formes linéaires que celles obtenues de cette façon. On va voir que c'est encore une propriété géométrique.

Théorème 1.19 (Riesz) *Toute forme linéaire (continue) a sur X peut s'écrire comme le produit scalaire par un élément fixe p de X. En outre, $\|a\| = \|p\|$.*

Démonstration Soit donc a une forme linéaire continue su X. Il faut montrer qu'il existe un vecteur $p \in X$ tel que $\forall x \in X$, $a(x) = (p, x)$. Si a est identiquement nulle, le problème est banal : on

prend $p = 0$. Supposons donc que a est non identiquement nulle. Le noyau K de a est un sous espace vectoriel, fermé parce que a est continue, non réduit à zéro puisque pour x_1 et x_2 non colinéaires d'image non nulle, $0 \neq a(x_2)x_1 - a(x_1)x_2 \in K$.

Prenons un x tel que $a(x) \neq 0$, et $y = x/a(x)$ de sorte que $a(y) = 1$, et appelons $\hat{y} = P_K(y)$ la projection de y sur K. Alors, nous savons que $y - \hat{y}$ est non nul et orthogonal à K, et donc aussi

$$p = \frac{1}{\|y - \hat{y}\|^2}(y - \hat{y})\,.$$

Insistons sur le fait que par définition $\hat{y} \in K$ de sorte que $a(\hat{y}) = 0$, ou $a(y - \hat{y}) = 1$.

Soit $x \in X$ quelconque. Nous affirmons que $P_K(x) = \hat{x}$ avec

$$\hat{x} = x - a(x)(y - \hat{y})\,.$$

En effet, l'expression ci-dessus donne par linéarité $a(\hat{x}) = a(x) - a(x)a(y - \hat{y}) = 0$. En outre $x - \hat{x}$ est parallèle à p donc orthogonal à K, ce qui caractérise bien $P_K(x)$.

Donc $(p, \hat{x}) = 0$, soit, puisque par construction $(p, y - \hat{y}) = 1$, $(p, x) = a(x)$. L'affirmation sur les normes découle de Cauchy-Schwarz. Le théorème est démontré.

Nous pouvons commenter un peu plus ce résultat. L'ensemble des formes linéaires sur X est appelé l'espace *dual* de X, souvent noté X'. Notre remarque liminaire indiquait donc qu'en dimension finie, le dual de $\mathbb{R}^n$ peut être *identifié* à $\mathbb{R}^n$ lui-même si on accepte de lire les n éléments de la matrice de la forme linéaire comme les coordonnées d'un vecteur. Donc en *transposant* cette matrice. Et la forme linéaire est bien le produit scalaire de ce vecteur obtenu par transposition et x. Nous remarquions ensuite qu'en dimension quelconque X peut être identifié au moins à un sous-ensemble de X' via le produit scalaire. Ce que nous dit le théorème de Riesz, dans ce langage, est que X peut être identifié à son dual.

Remarquons que X peut donc aussi être identifié au dual de son dual, appelé son *bidual*. les espaces qui ont cette propriété sont appelés "reflexifs".

Voici enfin un théorème qui mène à une conclusion topologique.

Théorème 1.20 *Tout sous-ensemble convexe fermé est égal à l'intersection de tous les demi-espaces fermés qui le contiennent.*

Démonstration Qu'il soit *contenu* dans cette intersection est évident. Mais comme pour tout point qui n'est pas dans le convexe on peut trouver un demi-espace qui contient le convexe et pas ce point, aucun point extérieur au convexe n'est dans cette intersection.

Ce théorème serait de peu d'intérêt si ce n'était pour son corollaire. Celui-ci concerne la topologie faible, dont il a été question dans le complément à la démonstration du théorème de séparation large. Rappelons qu'on dit que qu'une suite x_k tend *faiblement* vers $\bar{x}$ si pour tout $p \in X$, $(p, x_k) \to (p, \bar{x})$. Cette topologie ne présente d'intérêt qu'en dimension infinie, parce que les fermés bornés n'y sont plus compacts en topologie ordinaire —ou forte—, mais le sont encore en topologie faible. (Pour tous les espaces reflexifs, donc notamment les Hilbert. On voit le rôle du théorème de Riesz.)

Corollaire 1.21 *Les convexes fermés d'un espace de Hilbert sont faiblement fermés.*

Démonstration Remarquons d'abord que l'affirmation n'est pas banale. En effet, la convergence forte entraine la convergence faible mais pas le contraire, donc il y a "plus" de suites convergentes faibles que fortes, donc la propriété de contenir les limites de ses suites convergentes (caractère fermé) est plus contraignante en topologie faible qu'en topologie forte. Mais la définition même de la topologie faible implique banalement que les demi-espaces fermés sont fermés pour la topologie faible. Et la représentation du convexe fermé comme une intersection de fermés faibles établit son caractère faiblement fermé.

1.2 Fonctions convexes

Nous examinons maintenant des fonctions de X dans $\mathbb{R}$. Le meilleur moyen de visualiser la propriété de convexité est de se référer au graphe de la fonction, ce que nous ferons rapidement. Une fonction convexe a son graphe dont la concavité est tournée vers le haut.

1.2.1 Propriétés de base

Il sera utile dans la suite de considérer des *fonctions étendues* comme des fonctions de $\mathbb{R}^n$ dans $\mathbb{R} \cup \{\infty\}$. Ce dernier espace sera noté $\bar{\mathbb{R}}$.

Celà nous force à donner une définition préalable :

Définition 1.10 (Domaine) *On appelle* domaine *d'une fonction étendue l'ensemble sur lequel elle prend des valeurs finies :*

$$D(u) = \{x \in X \mid u(x) < \infty\}\,.$$

Définition 1.11 (Fonction convexe) *Une fonction u d'un sous-ensemble convexe $C \subset X$ dans $\bar{\mathbb{R}}$ est dite*

- convexe *si*

$$\forall x, y \in X\,,\ \forall \lambda \in [0,1]\,, \quad u(\lambda x + (1-\lambda)y) \leq \lambda u(x) + (1-\lambda)u(y)\,, \tag{1.9}$$

- strictement convexe *si l'inégalité dans (1.9) est stricte dès que $x \neq y$ et $\lambda \notin \{0,1\}$,*
- fortement convexe *s'il existe un nombre réel positif α tel que*

$$\forall x, y \in X\,,\ \forall \lambda \in [0,1]\,, \quad u(\lambda x + (1-\lambda)y) \leq \lambda u(x) + (1-\lambda)u(y) - \alpha\frac{\lambda(1-\lambda)}{2}\|x-y\|^2\,. \tag{1.10}$$

L'inégalité (1.9) dit que *le graphe de u est sous la corde*. En effet, le graphe de la fonction affine $\lambda x + (1-\lambda)y \mapsto \lambda u(x) + (1-\lambda)u(y)$ est la corde reliant les points $(x, u(x))$ et $(y, u(y))$ du graphe de u. L'inégalité (1.10) renforce cette propriété en demandant que le graphe de u soit même en dessous d'une parabole de paramètre $\alpha/2$ passant par ces points.

D'une fonction vérifiant l'inégalité (1.10) on dit qu'elle est *α-convexe*. Érigeons en remarque formelle la banalité suivante :

Remarque 1.4 *La propriété de convexité est équivalente à la 0-convexité.*

Ceci nous permettra d'énoncer des résultats et de faire des démonstrations pour l'α-convexité, et de trouver les propriétés équivalentes pour la convexité simple en mettant $\alpha = 0$ dans les résultats.

De même que pour les sous-ensembles, on peut énoncer la convexité en termes de combinaison convexe d'un nombre m de points. On aura :

Proposition 1.22 *Une fonction u d'un convexe C dans $\bar{\mathbb{R}}$ est convexe si et seulement si, quelques soient les points x_i, $i = 1, \ldots, m$ de C et les λ_i réels positifs ou nuls sommant à un, on a*

$$u(\sum_i \lambda_i x_i) \leq \sum_i \lambda_i u(x_i)\,. \tag{1.11}$$

Pour retrouver des propriétés géométriques, nous avons besoin du concept suivant :

Définition 1.12 (Épigraphe) *On appelle*

– épigraphe *de u, qu'on notera $\mathcal{E}(u)$, le sous ensemble de $X \times \mathbb{R}$ situé "au dessus du graphe" de u, c'est à dire défini par*

$$\mathcal{E}(u) = \left\{ \begin{pmatrix} x \\ r \end{pmatrix} \in X \times \mathbb{R} \mid r \geq u(x) \right\},$$

– épigraphe strict *de u le sous ensemble de $X \times \mathbb{R}$ défini par une inégalité stricte ci-dessus.*

Donnons tout de suite une définition équivalente de la convexité, en termes d'ensemble convexe :

Proposition 1.23 *Une fonction est convexe si et seulement si son épigraphe est (un sous-ensemble) convexe.*

On laisse le lecteur vérifier cette assertion essentielle (mais facile).

Ces deux définitions de la convexité seront utiles. Remarquons par exemple deux propriétés qui découlent immédiatement de la première. La première est une banalité :

Proposition 1.24 *Le domaine d'une fonction convexe est convexe.*

On peut en outre faire remarquer qu'une fonction convexe de $C \subset X$ peut toujours être étendue à tout X en posant $u(x) = \infty$ en dehors de C. Nous éviterons pourtant cet artifice.

La deuxième proposition, pour facile qu'elle soit, n'en est pas moins importante :

Proposition 1.25 *Si une fonction convexe atteint son minimum, l'ensemble des points où il est atteint est convexe. Si une fonction strictement convexe atteint son minimum, c'est en un point unique.*

La preuve est immédiate en utilisant (1.9).

Au contraire, le résultat suivant, tout aussi facile mais tout aussi important, est une conséquence immédiate de la caractérisation en termes d'épigraphe :

Proposition 1.26 *Le sup d'une famille (finie ou infinie) de fonctions convexes est convexe.*

En effet, l'épigraphe du sup d'une famille de fonctions est l'intersection des épigraphes des fonctions de la famille.

Enfin une propriété utile qui complète le lien entre fonctions convexes et ensembles convexes :

Proposition 1.27 *Si f est une fonction convexe de $C \subset X$ dans $\bar{\mathbb{R}}$, l'ensemble*

$$K = \{x \in C \mid f(x) \leq 0\}$$

est convexe.

La démonstration élémentaire est laissée au lecteur. En corollaire, notons qu'en conséquence, l'ensemble des x qui satisfonts m inégalités de cette forme est également convexe, comme intersection de m ensembles convexes.

1.2.2 Régularité des fonctions convexes

On démontre l'important résultat suivant, que nous énonçons dans une forme spécifique au cas de la dimension finie, c'est à dire pour une fonction u d'un sous-ensemble convexe C de $\mathbb{R}^n$ dans $\bar{\mathbb{R}}$. Nous verrons dans la démonstration ce qui s'étend aux cas où $C \subset X$ un espace de Hilbert (de dimension infinie).

Théorème 1.28 *Les fonctions convexes de $\mathbb{R}^n$ sont continues à l'intérieur de leur domaine.*

Démonstration La preuve utilise deux lemmes dont le premier, de nature géométrique, demeure en dimension infinie :

Lemme 1.29 *Une fonction convexe localement bornée au voisinage d'un point x de l'intérieur de son domaine y est continue.*

Preuve du lemme La preuve qui suit a un support géométrique qu'on renonce à indiquer ici, mais le lecteur est invité à le retrouver.

Soit x un point intérieur au domaine D de la fonction convexe u, $\varepsilon > 0$ le rayon d'une boule B (fermée) centrée en x et contenue dans D, et M un majorant de u sur cette boule. Soit y un point de B. On introduit les extrémités z et z' du diamètre de B passant par y :

$$z = x + \varepsilon \frac{y-x}{\|y-x\|}\,, \qquad z' = x - \varepsilon \frac{y-x}{\|y-x\|}\,.$$

On a donc $u(z) \leq M$ et $u(z') \leq M$. On a

$$x = \frac{\|y-x\|}{\varepsilon + \|y-x\|} z' + \frac{\varepsilon}{\varepsilon + \|y-x\|} y\,,$$

d'où, par la convexité de u

$$u(x) \leq \frac{\|y-x\|}{\varepsilon + \|y-x\|} M + \frac{\varepsilon}{\varepsilon + \|y-x\|} u(y)\,,$$

que nous ré-écrivons

$$u(y) \geq u(x) - \frac{\|y-x\|}{\varepsilon}(M - u(x))\,. \tag{1.12}$$

De même, on a

$$y = \frac{\|y-x\|}{\varepsilon} z + \left(1 - \frac{\|y-x\|}{\varepsilon}\right) x$$

d'où

$$u(y) \leq \frac{\|y-x\|}{\varepsilon} M + \left(1 - \frac{\|y-x\|}{\varepsilon}\right) u(x)$$

que nous ré-écrivons

$$u(y) \leq u(x) + \frac{\|y-x\|}{\varepsilon}(M - u(x))\,. \tag{1.13}$$

Les deux inégalités (1.12) et (1.13) établissent la continuité de u en x.

Et voici où nous utilisons la dimension de l'espace :

Lemme 1.30 *Une fonction convexe de $\mathbb{R}^n$ est localement bornée en tout point intérieur à son domaine.*

Preuve Soit u une fonction convexe d'un convexe C de $\mathbb{R}^n$ dans $\bar{\mathbb{R}}$ et D son domaine. Soit $x \in \overset{\circ}{D}$. On note qu'il existe un hypercube d'arrête positive dont x est le centre. Tout y de cet hypercube est (de plusieurs façons) une combinaison convexe de ses sommets x_k, $k = 1, \ldots, 2n$. Par la relation (1.11), on en déduit que $u(y) \leq \max_k u(x_k)$, ce qui établit le résultat.

Indiquons à titre de complément ce qui se passe si la dimension n'est pas finie :

Théorème 1.31 *Une fonction convexe localement bornée en un point intérieur de son domaine est continue sur tout l'intérieur de son domaine.*

Ceci est une conséquence du lemme 1.29 ci-dessus, et du suivant :

Lemme 1.32 *Une fonction convexe localement bornée en un point intérieur à son domaine est localement bornée en tout point intérieur de ce domaine.*

Preuve du lemme Soit donc u une fonction convexe d'un convexe $C \subset X$ dans $\bar{\mathbb{R}}$ et D son domaine. Soit $x \in \overset{\circ}{D}$, $B \subset \overset{\circ}{D}$ une boule de rayon $\varepsilon > 0$ centrée en x et M un majorant de $u(x)$ sur B. Soit aussi $y \in \overset{\circ}{C}$ un autre point de l'intérieur de D. Il existe un nombre $a > 0$ tel que la boule de rayon a centrée en y soit contenue dans D. Considérons le point $z = y + a(y-x)/\|y-x|$ qui appartient donc à D et $u(z) = N$. La boule de rayon $a\varepsilon/(\|y-x\|+a)$ centrée en y est homothétique de B dans une homothétie de centre z. Nous allons montrer que dans cette boule u est majorée par $\max\{M, N\}$, ce qui établira le lemme. Soit en effet y' dans cette boule, et $x' = (1 + \|y-x\|/a)y' - (\|y-x\|/a)z$ et remarquons que $x = (1+\|y-x\|/a)y - (\|y-x\|/a)z$, de sorte que $x'-x = (1+\|y-x\|/a)(y'-y)$ et que donc $\|x'-x\| \leq \varepsilon$ soit $x' \in B$ donc $u(x') \leq M$. Par construction, y' est une combinaison convexe de x' et z, précisément $y' = (a/(a+\|y-x\|))x' + (\|y-x\|/(a+\|y-x\|))z$. En conséquence, $u(y') \leq \max\{M, N\}$.

Il est utile de remarquer que si la fonction est continue, l'épigraphe est fermé et l'épigraphe strict ouvert (exercice). La réciproque est plus délicate. Nous donnons à titre de complément d'information le résultat qui est correct :

Proposition 1.33 *Une fonction u est*

- *semi-continue inférieurement (s.c.i) si et seulement si son épigraphe est fermé,*
- *semi-continue supérieurement (s.c.s) si et seulement si son épigraphe strict est ouvert.*

Rappelons qu'une fonction u est dite s.c.i. si, pour toute suite de points $\{x_k\}$ tendant vers un $\bar{x}$ on a

$$u(\bar{x}) \leq \liminf u(x_k)\,,$$

au contraire, la fonction est s.c.s. si

$$u(\bar{x}) \geq \limsup u(x_k)\,.$$

Une fonction s.c.i. et s.c.s. est donc continue.

On laisse en exercice de montrer la version fine du théorème de Weierstrass, qui explique pourquoi ce concept est important en optimisation :

Théorème 1.34 *Une fonction s.c.i. atteint son* $\min$ *sur un compact, une fonction s.c.s son* $\max$.

En conséquence, on a le théorème suivant, valide en dimension infinie :

Corollaire 1.35 *Une fonction convexe continue sur un fermé borné y atteint son minimum.*

Démonstration En effet, continue son épigraphe est fermé. Mais convexe fermé, cet épigraphe est aussi fermé en topologie faible, par le corollaire 1.21, donc la fonction est s.c.i. en topologie faible, topologie dans laquelle le fermé borné est compact.

1.2.3 Sous-différentiel

Nous avons indiqué en préliminaire que l'analyse convexe a été considérée comme le point de départ de l'analyse non différentiable. Le concept qui vient est une espèce de succédané de gradient, et est pour beaucoup dans cette appréciation.

Théorème 1.36 *Soit u une fonction convexe (continue) de $C \subset X$ dans $\bar{\mathbb{R}}$. En tout point intérieur à son domaine il existe au moins un vecteur p de X tel que*

$$\forall y \in C\,, \quad u(y) \geq u(x) + (p, y - x)\,. \tag{1.14}$$

Démonstration Soit x intérieur au domaine D de notre fonction convexe u. Le point $(x; u(x))$ de $C \times \mathbb{R}$ est sur la frontière de l'épigraphe de u. Comme u est continue, l'épigraphe strict est ouvert, et peut donc être séparé de son point frontière, au sens où il doit exister un élément $(q; \theta) \in C \times \mathbb{R}$ non nul tel que pour tout point $(y; r)$ de l'épigraphe strict, donc tel que $r > u(y)$, on ait (la première parenthèse de l'inégalité ci-dessous désigne bien un produit scalaire de X, dont la somme avec le produit des composantes dans $\mathbb{R}$ est le produit scalaire de $X \times \mathbb{R}$)

$$(q, y - x) + \theta(r - u(x)) < 0\,.$$

Notons d'abord que θ est nécessairement négatif. En effet, il ne saurait être positif, car l'inégalité ci-dessus doit être satisfaite pour tout $r > u(y)$, donc on peut y faire tendre r vers $+\infty$. Mais θ ne peut non plus être nul parce que x étant intérieur à C, on n'a surement pas $(q, y - x) < 0$ pour tout $y \in C$. Posons donc

$$p = \frac{1}{|\theta|} q\,.$$

L'inégalité ci-dessus devient (après division par $|\theta|$) :

$$\forall r > u(y)\,, \quad (p, y - x) < r - u(x)\,.$$

Il suffit de faire tendre r vers $u(y)$ par valeurs supérieures dans cette inégalité, et le théorème en découle.

On introduit alors une définition, qui subsiste que u soit continue ou non

Définition 1.13 (Sous-différentiel) *On appelle sous-différentiel de u en x, et on note $\partial u(x)$, l'ensemble des p de $\mathbb{R}^n$ qui satisfont (1.14).*

Le théorème 1.36 se lit alors : en tout point où une fonction convexe est continue (en dimension finie, en tout point intérieur à son domaine) son sous-différentiel est non vide.

Une propriété facile est

Proposition 1.37 *Le sous différentiel est convexe fermé en tout x.*

Et une autre banalité mérite d'être soulignée :

Théorème 1.38 *Une fonction convexe atteint son minimum en un point x^* de son domaine si et seulement si $0 \in \partial u(x^*)$.*

L'α-convexité permet de renforcer l'inégalité (1.14) comme suit :

Théorème 1.39 *Pour une fonction u α-convexe sur C, on a :*

$$\forall p \in \partial u(x)\,,\ \forall y \in C\,,\quad u(y) \geq u(x) + (p, y-x) + \frac{\alpha}{2}\|y-x\|^2\,. \tag{1.15}$$

Démonstration L'inégalité d'α-convexité s'écrit, après division par λ,

$$u(y) \geq \frac{1}{\lambda}u((1-\lambda)x + \lambda y) - \frac{1-\lambda}{\lambda}u(x) + \alpha\frac{1-\lambda}{2}\|y-x\|^2\,.$$

En outre, par l'inégalité du sous-différentiel

$$u((1-\lambda)x + \lambda y) \geq u(x) + \lambda(p, y-x)\,,$$

de sorte que l'inégalité précedente donne

$$u(y) \geq u(x) + (p, y-x) + \alpha\frac{1-\lambda}{2}\|y-x\|^2\,.$$

Il suffit de faire tendre λ vers 0 dans cette dernière inégalité pour obtenir le résultat annoncé.

En particulier, en utilisant le théorème 1.38, on voit que

Corollaire 1.40 *Si u α-convexe atteint son minimum en x^*, alors*

$$\forall x \in C\,,\quad u(x) \geq u(x^*) + \frac{\alpha}{2}\|x-x^*\|^2 \tag{1.16}$$

On voit aussi facilement que le sous-différentiel est un opérateur *monotone*, c'est à dire que (rappelons que le cas $\alpha = 0$ donne la convexité simple)

Théorème 1.41 *Pour une fonction u α-convexe de C dans $\bar{\mathbb{R}}$, on a*

$$\forall x, y \in X\,,\ \forall p \in \partial u(x)\,,\ \forall q \in \partial u(y)\,,\quad (p-q, x-y) \geq \alpha\|x-y\|^2\,. \tag{1.17}$$

Démonstration Il suffit décrire (1.15) en x et en y et d'ajouter.

À nouveau, grâce au théorème 1.38, on a

Corollaire 1.42 *Si u α-convexe atteint son minimum en x^*, alors*

$$\forall x \in \overset{\circ}{C}\,,\ \forall p \in \partial u(x)\,,\quad \|p\| \geq \alpha\|x-x^*\| \tag{1.18}$$

Démonstration On a en effet $0 \in \partial u(x^*)$, donc

$$(p, x-x^*) \geq \alpha\|x-x^*\|^2\,,$$

et en utilisant l'inégalité de Cauchy-Schwarz l'inégalité annoncée.

Indiquons enfin une variante de l'inégalité d'Euler, mais cette fois sous la forme d'une condition *suffisante*, et concernant un minimum atteint en un point quelconque de C (il étend donc le théorème 1.38) :

Théorème 1.43 *Soit u une fonction convexe de C dans $\bar{\mathbb{R}}$. Si en $\bar{x}$, il existe $p \in \partial u(x)$ tel que $-p$ soit une normale extérieure à C, alors u atteint son minimum sur C en $\bar{x}$.*

Démonstration Il suffit de mettre ensemble les inégalités (1.14) et celle de la définition 1.16.

Le caractère nécessaire de cette affirmation est une question un peu plus délicate que nous n'examinerons pas.

1.2.4 Dérivées et convexité

Dérivées et dérivées partielles

Soit une fonction $u(\cdot) : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ d'un ouvert Ω de $\mathbb{R}^n$ dans $\mathbb{R}$. Le lecteur connaît la notion de dérivée partielle. Formellement, on peut écrire que $\partial u/\partial x_i$ est la dérivée de l'application partielle

$$x_i \mapsto u(x_1, x_2, \ldots, x_i, \ldots, x_n).$$

La notation $\dfrac{\partial u}{\partial x_i}$, que nous utiliserons, présente un gros inconvénient qu'il faut souligner ici. C'est l'usage du *nom donné à l'argument* (ici x_i) dans le *nom de la fonction* dérivée partielle. Ainsi, comment doit-on écrire l'accroissement au premier ordre de u entre les points

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} y_1 + z \\ y_2 \\ \vdots \\ y_n \end{pmatrix} ?$$

Nous le noterons (en appelant e_1 le vecteur $(1,\, 0,\, \ldots,\, 0)^t$)

$$u(y + ze_1) = u(y) + \frac{\partial u}{\partial x_1}(y)z + o(z)\,,$$

et surtout pas

$$u(y + ze_1) = u(y) + \frac{\partial u}{\partial y_1}(y)z + o(z)\,.$$

Car on ne saurait changer le nom d'une fonction (ici la dérivée partielle par rapport à la première variable) à chaque fois qu'on change le nom de son argument. Si on faisait ainsi, que deviendrait cette dérivée partielle au point $(1,\, 1,\, \ldots,\, 1)^t$?

Pour cette raison, Dieudonné propose de noter les dérivées partielles $D_i u(\cdot)$ pour la dérivée par rapport à la variable de rang i. En cas de besoin, on invite le lecteur à avoir recours à cette notation qui évite les ambigüités.

Nous placerons d'habitude les vecteurs en colonne, et les dérivées partielles par rapport aux coordonnées d'un vecteur en ligne, réservant la notation $u'(x)$ à cette présentation :

$$u'(x) = \begin{pmatrix} \frac{\partial u}{\partial x_1} & \frac{\partial u}{\partial x_2} & \cdots & \frac{\partial u}{\partial x_n} \end{pmatrix}.$$

Ainsi nous aurons, utilisant un produit matriciel ordinaire, la formule fondamentale préservée :

$$u(x + h) = u(x) + u'(x)h + o(\|h\|)\,. \tag{1.19}$$

Nous utiliserons la notation ∇u pour désigner le vecteur colonne des dérivées partielles, donc le transposé de u'. Ainsi (1.19) s'écrit aussi

$$u(x + h) = u(x) + (\nabla u(x), h) + o(\|h\|)\,,$$

où (y, z) désigne le produit scalaire des vecteurs y et z.

Si la fonction u est elle même vectorielle :

$$u(x) = \begin{pmatrix} u_1(x) \\ u_2(x) \\ \vdots \\ u_m(x) \end{pmatrix},$$

on fera des $u_i'(x)$ les lignes de la matrice de type $m \times n$:

$$u' = \begin{pmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \cdots & \frac{\partial u_1}{\partial x_n} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \cdots & \frac{\partial u_2}{\partial x_n} \\ & \vdots & & \\ \frac{\partial u_m}{\partial x_1} & \frac{\partial u_m}{\partial x_2} & \cdots & \frac{\partial u_m}{\partial x_n} \end{pmatrix},$$

et la formule fondamentale (1.19) reste correcte.

Dans la notation de Dieudonné,

$$u' = \begin{pmatrix} D_1u_1 & D_2u_1 & \cdots & D_nu_1 \\ D_1u_2 & D_2u_2 & \cdots & D_nu_2 \\ & \vdots & & \\ D_1u_m & D_2u_m & \cdots & D_nu_m \end{pmatrix}.$$

Dérivation en chaîne et dérivées directionnelles

Dérivées en chaîne Avec les choix de notation ci-dessus, on préserve aussi la formule des "dérivées en chaîne" : si $x = v(y)$ et $w(y) = u(x) = u(v(y))$, avec u et v continuement dérivables (on dit "de classe C^1),

$$w'(y) = u'(v(y))v'(y)$$

quelques soient les dimensions des vecteurs en cause. Par exemple, Si $v : \mathbb{R}^p \to \mathbb{R}^n$ et $u : \mathbb{R}^n \to \mathbb{R}^m$, alors u' est de type $m \times n$ et v' de type $n \times p$, de sorte que le produit matriciel peut bien être fait, et w' est de type $m \times p$. Tout ceci découle nécessairement de la formule (1.19).

Dérivée directionnelle Appliquons cela à la dérivée de la "coupe" d'une fonction le long d'une droite. Soit $\xi(t) = x + th$ une droite de $\mathbb{R}^n$. Ici, x et h sont fixés dans $\mathbb{R}^n$, et t varie dans $\mathbb{R}$. Posons $U(t) = u(\xi(t))$. C'est donc la restriction de la fonction u à la droite de direction h passant par x. On a en appliquant les règles ci-dessus :

$$U'(t) = u'(\xi(t))h\,,$$

et nous l'utiliserons surtout en $t = 0$:

$$U'(0) = (\nabla u(x), h)\,. \tag{1.20}$$

Exercice 1.1 (important) *: En déduire que*

- *la* ligne de plus grande pente *est parallèle à ∇u (et opposée pour descendre),*
- *cette direction est* orthogonale à la courbe de niveau *qui est la courbe* $\{y \mid u(y) = u(x)\}$.

.

Deuxième ordre Rappelons enfin la forme du développement au deuxième ordre d'une fonction scalaire de n variables :

$$u(x+h) = u(x) + u'(x)h + \frac{1}{2}h^t D^2u(x)h + o(\|h\|^2)\,, \tag{1.21}$$

ou encore, avec le reste de Lagrange

$$u(x+h) = u(x) + u'(x)h + \frac{1}{2}h^t D^2u(x+\theta h)h\,, \tag{1.22}$$

pour un $\theta \in [0,1]$.

Comme toujours, ces formules se déduisent de celles du cas scalaire en regardant la fonction $U(t) = u(x+th)$.

Dérivées des fonctions convexes

Bien que la convexité soit une théorie fondamentalement "non différentiable", les fonction convexes une ou deux fois dérivables ont des propriétés importantes. Les premières se déduisent du fait suivant :

Théorème 1.44 *Si la fonction convexe u de $C \subset X$ est dérivable en $x \in \overset{\circ}{C}$, alors $\partial u(x) = \{\nabla u(x)\}$.*

Démonstration Aussi simple que soit ce théorème, nous en indiquons la démonstration : prenons un $p \neq \nabla u(x)$, et $h \in X$ tel que $(p,h) > (\nabla u(x), h)$. Alors, nous savons que $u(x+th) = u(x) + t(\nabla u(x), h) + o(t)$, de sorte que pour t suffisamment petit positif, on aura $u(x+th) < u(x) + (p, th)$, donc $p \notin \partial u(x)$.

Une conséquence importante, quoique facile, est la suivante :

Théorème 1.45 *Soit u une fonction convexe dérivable de C dans $\mathbb{R}$. Elle atteint son minimum en $x^* \in C$ si et seulement si*

$$\forall x \in C\,, \quad (u'(x^*), x - x^*) \geq 0\,. \tag{1.23}$$

En reconnaissant dans le gradient un élément du sous-différentiel, de (1.15) et (1.17), on obtient

Théorème 1.46 *Soit u une fonction α-convexe dérivable de $C \subset X$, on a*

$$\forall x, y \in C\,, \quad u(y) \geq u(x) + u'(x)(y-x) + \frac{\alpha}{2}\|y-x\|^2\,, \tag{1.24}$$

$$\forall x, y \in C\,, \quad (\nabla u(x) - \nabla u(y), x - y) \geq \alpha\|x-y\|^2\,. \tag{1.25}$$

Enfin, la deuxième inégalité ci-dessus permet de montrer le caractère nécessaire de l'affirmation suivante :

Théorème 1.47 *Une fonction u de C convexe de $\mathbb{R}^n$ deux fois continuement dérivable est α-convexe si et seulement si*

$$\forall x \in \overset{\circ}{C}\,, \quad D^2u(x) \geq \alpha I \tag{1.26}$$

où l'inégalité est bien sûr au sens des matrices définies positives.

Démonstration Nous avons déjà indiqué que le caractère nécessaire découle facilement de l'inégalité (1.25). Le caractère suffisant s'obtient en évaluant

$$u'(x+\theta h)h = u'(x)h + \int_0^\theta h^t D^2u(x+th)h\,dt \geq u'(x)h + \alpha\|h\|^2\theta$$

puis

$$u(x+h) = u(x) + \int_0^1 u'(x+\theta h)h\,d\theta \geq u(x) + u'(x)h + \frac{\alpha}{2}\|h\|^2\,,$$

On utilise cette estimation de la facon suivante. On se met en $x_\lambda = \lambda x + (1-\lambda)y$, on écrit $x = x_\lambda + (1-\lambda)(x-y)$ et $y = x_\lambda - \lambda(x-y)$, et on utilise la formule ci-dessus avec $h = (1-\lambda)(x-y)$ puis avec $h = -\lambda(x-y)$. Cela donne

$$\begin{aligned} u(x) &\geq u(x_\lambda) + (1-\lambda)u'(x_\lambda)(x-y) + \frac{\alpha}{2}(1-\lambda)^2\|x-y\|^2\,, \\ u(y) &\geq u(x_\lambda) - \lambda u'(x_\lambda)(x-y) + \frac{\alpha}{2}\lambda^2\|x-y\|^2\,. \end{aligned}$$

En multipliant la première inégalité par λ, la seconde par $(1-\lambda)$ et en ajoutant, il vient exactement l'inégalité d'α-convexité, ce qu'il fallait démontrer.

Complément

À titre de complément mathématique, indiquons la relation plus fine qui lie le sous différentiel d'une fonction convexe et ses dérivées.

Étant donné un convexe $D \subset X$, on appelle *fonction support* de D la fonction $\sigma_D(p)$ de X dans $\mathbb{R}$ définie par

$$\sigma_D(p) = \sup_{x\in D}(p,x)\,.$$

On montre facilement que σ_D est une fonction convexe, et que

$$\bar{D} = \{x \in X \mid \forall p \in X,\ (p,x) \leq \sigma_D(p)\}\,.$$

Donc la fonction support caractérise complètement un convexe fermé.

Étant donnée une fonction u de X dans $\mathbb{R}$, on appelle *dérivée directionnelle dans la direction* h, notée $Du(x;h)$, la dérivée à droite en zéro de la fonction de $\mathbb{R}$ dans $\mathbb{R}$ définie par $t \mapsto u(x+th)$.

Il est facile de voir que pour une fonction convexe, le quotient différentiel

$$\frac{u(x+th)-u(x)}{t}\,,$$

la pente de la corde du graphe de u entre x et $x+th$, est croissant avec t, donc décroit quand t décroit vers 0, et est minorée par la pente d'une corde pour $t < 0$. Donc la dérivée à droite existe et est finie.

On vérifie le fait suivant :

Proposition 1.48 *En tout point intérieur à son domaine,une fonction convexe continue admet des dérivées directionnelles dans toutes les directions, et*

$$Du(x;h) = \sigma_{\partial u(x)}(h)\,.$$

Ainsi, le sous-différentiel et l'ensemble des dérivées directionnelles sont deux données équivalentes. Cette propriété peut servir à compléter létude des conditions nécessaires d'optimalité.

Indiquons enfin une conséquence facile, qui concerne la dérivée de u —sa dérivée de Gâteaux si X est de dimension infinie— et qui précise la proposition 1.44 :

Proposition 1.49 *Une fonction convexe continue est dérivable en x si et seulement si son sous différentiel en x est un singleton.*

1.3 Optimisation sous contraintes

1.3.1 Le théorème de Kuhn et Tucker

On va utiliser l'arsenal développé ici pour établir un des théorèmes fondamentaux de l'analyse convexe et de l'optimisation sous contraintes, d'où nous déduirons un algorithme important dans le chapitre consacré aux algorithmes.

Le problème considéré

Le problème standard que nous considérons ici est défini par une fonction à minimiser et des contraintes.

La fonction à minimiser est une fonction convexe u de $\mathbb{R}^n$ dans $\mathbb{R}$, ou d'un sous-ensemble convexe $C \subset \mathbb{R}^n$ dans $\mathbb{R}$. Mais la question du domaine de définition de u sera reprise avec les contraintes.

Si nous souhaitons bien nous intéresser au cas d'une fonction de n variables réelles, il n'en reste pas moins que toute la théorie de l'analyse convexe ne doit rien à la dimension finie. Aussi nous appellerons comme ci-dessus X l'espace où vit la variable d'optimisation x, et le lecteur intéressé pourra lire la suite en pensant que X est un espace de Hilbert quelconque.

Dans les'algorithmes du chapitre 3, u sera toujours supposée dérivable (mais la notion de gradient demeure dans un Hilbert quelconque) et généralement fortement convexe pour que les algorithmes convergent. Nous n'avons besoin d'aucune de ces hypothèses ici.

Examinons maintenant les contraintes, qui font l'intérêt de ce paragraphe. L'ensemble des x admissibles sera un sous-ensemble convexe, mais nous distinguons deux façons de le spécifier, deux familles de contraintes.

D'une part, les contraintes que nous ne *dualiserons* pas, qui restent exprimées de façon que nous qualifierons d'*abstraite* par $x \in C$ pour un certain convexe C dont on ne dit pas plus comment il est spécifié. D'autre-part des contraintes que nous dualiserons, spécifiées de manière *concrête* à l'aide de m fonctions $f_i(\cdot)$ de $\mathbb{R}^n$ (ou simplement C) dans $\mathbb{R}$, par les inégalités

$$f_i(x) \leq 0\,, \quad i = 1, \ldots, m\,. \tag{1.27}$$

Deux remarques à ce propos.

1. En application de la proposition 1.27 (et du commentaire qui la suit), l'ensemble des x de C qui satisfont ces contraintes est bien convexe.
2. Le choix des contraintes qu'on traite de façon concrête ou abstraite est généralement offert à l'utilisateur. Son choix peut être guidé par les algorithmes qu'il entend mettre en œuvre.

On peut écrire les contraintes (1.27) sous la forme

$$f(x) \leq 0 \tag{1.28}$$

pourvu qu'on convienne de la signification de $f \leq 0$ dans $\mathbb{R}^m$ comme voulant dire $f_i \leq 0$, $i = 1, \ldots, m$. C'est ce que nous ferons par la suite, utilisant aussi la notation $p \geq 0$ pour un vecteur p de $\mathbb{R}^m$ avec la signification évidente.

De même qu'on a appelé X l'espace où vit x, appelons F celui où vit $f(x)$. On a en fait défini une relation d'ordre partiel sur F, et (1.28) ne fait que l'utiliser. On aurait pu utiliser une autre relation d'ordre partiel, définie de la façon classique pour un espace vectoriel. On choisit un *cone* —c'est à dire un sous-ensemble invariant par multiplication par un scalaire positif— convexe fermé qu'on appelle P. Alors, $f \geq 0$ si $f \in P$, bien-sûr $f \leq 0$ si $-f \geq 0$, et $f > 0$ si $f \in \overset{\circ}{P}$ et $f < 0$ si $-f \in \overset{\circ}{P}$. Si on fait ainsi, on ne peut plus identifier F à son dual. Il faut en effet appeler p du dual de F, non-négatif : $p \geq 0$, si pour tout $f \geq 0$ on a $(p, f) \geq 0$. On vérifie facilement que l'ensemble des éléments positifs du dual est encore un cone convexe fermé, dit *cone polaire* de P.

Le cas (1.27) reste notre référence, mais on peut lire toute la suite avec une relation d'ordre plus générale, et même avec un espace F de dimension infinie. On peut écrire la convexité de f en termes de cette relation d'ordre. Le caractère convexe de l'ensemble (1.28) demeure grâce au caractère convexe de P (exercice). Seule la discussion de la signification détaillée de la condition des écarts complémentaires ci-dessous sera spécifique au cas de référence.

Théorème de Kuhn-Tucker

On rappelle que les fonctions u et f sont supposées convexes (continues). On posera

$$K = C \cap \{x \mid f(x) \leq 0\}\,. \tag{1.29}$$

On appelle problème $\mathcal{P}$ le problème

$$\min_{x \in K} u(x)\,. \tag{1.30}$$

On suposera en outre que pour tout réel r, l'ensemble $\{x \in K \mid u(x) \leq r\}$ est fermé borné.

On fait enfin l'hypothèse de *qualification* des contraintes dite de Slater :

Hypothèse Il existe un $x_0 \in C$ tel que $f(x_0) < 0$, c'est à dire (dans le cas de référence) que toutes les $f_i(x_0)$ sont strictement négatives.

On peut affaiblir cette hypothèse en supposant qu'en x_0 une certain nombre des f_i *affines* sont nulles, et les autres strictement négatives. On laissera le lecteur faire cette extension, importante pour la programmation linéaire.

Introduisons le *Lagrangien* du problème $\mathcal{P}$. C'est une fonction de $X \times F'$ dans $\mathbb{R}$ (où F' désigne le dual de F, $\mathbb{R}^m$ dans le cas de référence) :

$$\mathcal{L}(x, p) = u(x) + (p, f(x))\,. \tag{1.31}$$

Théorème 1.50 *Le problème $\mathcal{P}$ admet (au moins) une solution x^*. En toute solution il existe un vecteur du dual de F $p^* \geq 0$ tel que l'inégalité de point-selle ci-dessous soit satisfaite :*

$$\forall x \in C\,, \forall p \geq 0\,, \quad \mathcal{L}(x^*, p) \leq \mathcal{L}(x^*, p^*) \leq \mathcal{L}(x, p^*)\,, \tag{1.32}$$

et en outre,

$$(p^*, f(x^*)) = 0\,. \tag{1.33}$$

Réciproquement, si $x^* \in C$ et $p^* \geq 0$ satisfont (1.32), alors x^* est optimal (en particulier $x^* \in K$) et satisfait (1.33).

Les variables p sont appelées les variables duales, et les p_i^* les *multiplicateurs de Kuhn et Tucker* associés aux contraintes correspondantes. Les contraintes "concrêtes" (1.27) ont été *dualisées* : l'inégalité de droite de (1.32) se lit comme un problème d'optimisation en x sous les seules contraintes $x \in C$.

Démonstration La démonstration va reposer sur une idée fondamentale de l'analyse : on va *perturber* le problème $\mathcal{P}$, et étudier l'influence de cette perturbation sur le résultat. Soit donc, pour un élément $a \in F$,

$$K(a) = \{x \in C \mid f(x) - a \leq 0\}\,.$$

Nous allons étudier la fonction

$$V(a) = \min_{x \in K(a)} u(x)\,,$$

et appelerons $\mathcal{P}(a)$ ce problème d'optimisation. Notons que le problème a bien une solution par la compacité supposée des ensembles de niveau et la continuité de u (faibles si nécessaire), et l'hypothèse de Slater qui nous affirme au moins que K n'est pas vide. Notons aussi que $K = K(0)$ et $u(x^*) = V(0)$.

Nous allons procéder à 9 affirmations successives :

1. La fonction V est convexe. Étudier la convexité de V c'est étudier $V(\lambda a + (1-\lambda)b)$. Soient donc $x_a \in K(a)$ et $x_b \in K(b)$ des solutions de $\mathcal{P}(a)$ et $\mathcal{P}(b)$ respectivement. Le point important est le suivant : par la convexité de f, on voit facilement que

$$\lambda x_a + (1-\lambda)x_b \in K(\lambda a + (1-\lambda)b)\,,$$

et donc, par un argument récurrent dans cette démonstration, que

$$u(\lambda x_a + (1-\lambda)x_b) \geq V(\lambda a + (1-\lambda)b)\,.$$

Par définition, $u(x_a) = V(a)$ et $u(x_b) = V(b)$, de sorte que par la convexité de u on a bien

$$\lambda V(a) + (1-\lambda)V(b) \geq u(\lambda x_a + (1-\lambda)x_b) \geq V(\lambda a + (1-\lambda)b)\,.$$

2. La fonction V est continue en 0, et y a donc un sous-différentiel non vide. C'est ici qu'est réllement utilisée l'hypothèse de Slater : elle garantit que pour a dans un voisinage de 0, ce même $x_0 \in K(a)$, de sorte que

$$u(x_0) \leq V(a)\,.$$

Donc V est localement bornée au voisinage de 0.

3. Soit $p^* \in F'$ tel que $-p^* \in \partial V(0)$. Nous affirmons que $p^* \geq 0$. Soit en effet $a \geq 0$ dans F. Par définition du sous-différentiel on a

$$V(a) \geq V(0) - (p^*, a)\,.$$

Mais comme $a \geq 0$, on a $K(a) \supset K(0)$, et donc $V(a) \leq V(0)$. Donc $(p^*, a) \geq 0$ pour tout $a \geq 0$. Si la positivité à bien été définie comme en (1.27), il suffit de faire $a_i > 0$ et tous les autres a_j nuls pour en déduire que $p_i^* \geq 0$. Ceci pour tout i. (Dans le cas d'une relation d'ordre plus générale, la propriété précédente est la *définition* de $p^* \geq 0$.)

4. On a

$$\forall x \in C\,, \quad u(x^*) \leq u(x) + (p^*, f(x))\,. \tag{1.34}$$

Notons que trivialement, pour tout $x \in C$, $x \in K(f(x))$. Donc

$$\forall x \in C\,, \quad u(x) \geq V(f(x))\,.$$

En utilisant la définition du sous-différentiel

$$V(f(x)) \geq V(0) - (p^*, f(x))\,.$$

En mettant ensemble ces inégalités et la remarque faite plus haut que $V(0) = u(x^*)$, il vient l'inégalité annoncée.

5. On a l'égalité des *écarts complémentaires* (1.33) : $(p^*, f(x^*)) = 0$. Mettons en effet x^* pour x dans (1.34). Il vient $(p^*, f(x^*)) \geq 0$. Mais par définition, $x^* \in K(0)$ de sorte que $f(x^*) \leq 0$. Et nous avons montré que $p^* \geq 0$. Donc $(p^*, f(x^*)) \leq 0$. Ces deux inégalités confrontées donnent le résultat.

6. On a donc le caractère nécessaire de (1.32). En effet, d'une part, on peut rajouter $(p^*, f(x^*))$ au terme de gauche de (1.34) donnant l'inégalité de droite de (1.32). D'autre-part, il découle bien de (1.33) et de $f(x^*) \leq 0$ que pour tout $p \geq 0$, $(p, f(x^*)) \leq 0$, donnant l'inégalité de gauche de (1.32).

Réciproquement, supposons (1.32) satisfaite par un $x^* \in C$ et un $p^* \geq 0$.

7. En conséquence de l'inégalité de gauche de (1.32), on a (1.33). En effet, prenons successivement $p = 2p^*$ et $p = (1/2)p^*$, qui sont tous les deux non-négatifs. On en conclu successivement que $(p^*, f(x^*)) \leq 0$ et que $(p^*, f(x^*)) \geq 0$.

8. L'inégalité de gauche de (1.32) implique aussi que $f(x^*) \leq 0$ (donc $x \in K$). En effet, pour tout $p \geq 0$ on doit avoir $(p, f(x^*)) \leq 0$. Si un des $f_i(x^*)$ était positif, on prendrait cette coordonnée de p égale à 1 et toutes les autres nulles, contredisant l'inégalité. (Dans le cas d'une relation d'ordre plus générale, ceci vient du fait que le cone dual du cone positif de F' est bien le cone positif de F.)

9. x^* est la solution du problème $\mathcal{P}$. En effet, sachant que $(p^*, f(x^*)) = 0$, et en se souvenant que $p^* \geq 0$, il sufit de mettre un $x \in K$ dans l'inégalité de droite pour conclure $u(x^*) \leq u(x)$.
Le théorème est démontré.

Une remarque importante porte sur la condition des écarts complémentaires (1.33). Elle dit que pour chaque f_i qui n'est pas nulle en x^*, le multiplicateur p_i^* associé est nul. En effet, si $f_i(x^*) < 0$, localement la contrainte i n'intervient pas, elle n'est pas limitante, il est donc logique qu'elle n'intervienne pas dans la condition nécessaire. Elle est absente du lagrangien.

Insistons aussi sur l'interprétation "économique" des variables duales optimales, les multiplicateurs de Kuhn et Tucker. Puisque $-p^* \in \partial V(0)$, c'est une mesure de la *sensibilité du résultat optimal au niveau des contraintes*. Si u est un coût encouru, $f_i(x)$ à une constante près la quantité d'une denrée i consommée par la décision x, et $f_i(x) \leq 0$ l'expression du fait que la denrée i est disponible en quantité limitée, (c'est une denrée rare), alors p_i^* représente le prix unitaire maximum auquel on est prêt à acheter un petit surcroit de cette denrée pour diminuer ses coûts. Et alors, le lagrangien apparaît comme un coût total encouru si on nous faisait payer chaque denrée rare à ce prix.

On obtient facilement le corollaire suivant. (Remarquons que la notation $(p, f'(x))$ est abusive, car $f'(x)$ n'est pas un élément de X, mais de $\mathcal{L}(X, F)$, une matrice $m \times n$ dans le cas de référence. Il faut donc comprendre $(p, f'(x))$ comme l'application de X dans $\mathbb{R}$: $h \mapsto (p, f'(x)h)$, ou comme l'expression matricielle $p^t f'(x)$.)

Corollaire 1.51 *Si en outre des hypothèses faites, u et f sont dérivables, alors une condition nécessaire pour que $x^* \in K$ soit optimal est qu'il existe un vecteur $p^* \geq 0$ tel que*

$$u'(x^*) + (p^*, f'(x^*)) = 0\,.$$

Si en outre $(p^*, f(x^*)) = 0$, la condition est aussi suffisante.

La condition nécessaire est évidente. Pour la condition suffisante on rappelle seulement que comme $p^* \geq 0$, le lagrangien est convexe en x, donc la condition de dérivée nulle implpique que le lagrangien est minimisé en x^*.

1.3.2 Le théorème de Lagrange

Position du problème, conditions du premier ordre

Pour terminer, nous examinons le cas de contraintes égalité. Nous abandonnons toute hypothèse de convexité. Nous supposons au contraire que u est continuement dérivable, de même que les m fonctions scalaire f_i. Le problème considéré est de minimiser u sous les contraintes $f(x) = 0$.

Le théorème suivant est dû à Lagrange (et donc plus ancien que l'analyse convexe !)

Théorème 1.52 *Si x^* est une solution du problème posé, et si $f'(x^*)$ est surjective, alors il existe un vecteur λ de $\mathbb{R}^m$ tel que*

$$u'(x^*) + (\lambda, f'(x^*)) = 0\,. \tag{1.35}$$

(Pour la notation abusive $(\lambda, f'(x^*))$, voir la remarque précédent le corollaire 1.51 ci-dessus.)

Ce théorème dépend traditionnellement du théorème des fonctions implicites. Cette preuve classique s'étend en dimension infinie aussi, avec quelques précautions. (Nous ne le ferons pas ici.) Signalons pourtant que l'analyse non lisse moderne permet de donner une preuve dans le même esprit que la preuve ci-dessus du théorème de Kuhn et Tucker. On s'affranchit alors du "gros" théorème des fonctions implicites. Nous donnons pourtant ci-dessous la preuve traditionnelle.

Démonstration Supposons donc que $f'(x^*)$ est surjective. (Ce qu'on appelle encore la condition de qualification.) Il existe donc $m(< n)$ colonnes de $f'(x^*)$ linéairement indépendantes. Imaginons un moment que nous avons modifié la numérotation des coordonnées de x dans $\mathbb{R}^n$ pour regrouper en tête m d'entre-elles correspondant à m colonnes indépendantes de $f'(x^*)$. (Insistons sur le fait qu'il s'agit là d'une démarche conceptuelle, qui ne sera pas nécessaire pour appliquer le théorème.) Nous découpons la matrice $f'(x) = [f'_1\ f'_2]$ pour exhiber les m colonnes indépendantes dans la sous-matrice $f'_1(x^*)$, et x conformément, en

$$f'(x)x = \begin{bmatrix} f'_1 & f'_2 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = f'_1 x_1 + f'_2 x_2\,.$$

La matrice f'_1 évaluée en x^* est inversible, et le demeure par continuité dans un voisinage de x^*. Par le théorème des fonctions implicites, la contrainte $f(x) = 0$, qui est satisfaite en x^*, définit dans ce voisinage une fonction implicite $x_1 = \varphi(x_2)$, dérivable et de dérivée

$$\varphi'(x_2) = -(f'_1)^{-1} f'_2\,. \tag{1.36}$$

Pour des x satisfaisant la contrainte, le critère u s'écrit donc (avec un tout petit abus de notations)

$$u(x) = u(x_1, x_2) = u(\varphi(x_2), x_2)\,.$$

Nous noterons aussi u'_1 et u'_2 les dérivées de u par rapport à x_1 er x_2 respectivement. Maintenant, x_2 varie librement dans un voisinage ouvert de sa valeur x_2^* en x^*. Donc en un minimum, on doit avoir

$$u'_1(x^*)\varphi'(x_2^*) + u'_2(x^*) = 0\,,$$

soit, en tenant compte de (1.36),

$$-u_1'(x^*)f_1'(x^*)^{-1}f_2'(x^*) + u_2'(x^*) = 0\,. \tag{1.37}$$

Décidons de noter

$$\lambda^t := -u_1'(x^*)f_1'(x^*)^{-1}\,,$$

soit encore

$$u_1'(x^*) + \lambda^t f_1'(x^*) = 0\,, \tag{1.38}$$

L'égalité (1.37) devient

$$u_2'(x^*) + \lambda^t f_2'(x^*) = 0\,,$$

qui avec l'égalité précédente donne exactement (1.35).

Pour utiliser ce théorème, on cherche donc à calculer les $n+m$ inconnues x (n inconnues) et λ (m inconnues) à l'aide des $n+m$ équations (1.35) (n équations) et $f(x^*) = 0$ (m équations). Il est classique, dans une solution "à la main" de procéder par élimination en utilisant (1.35) pour calculer x^* en fonction de λ, puis de reporter dans $f = 0$ pour calculer λ, qu'on reporte enfin dans x^*. Mais cette démarche n'est pas toujours celle qui s'impose. Et nous serons plus intéressé par les méthodes numériques que nous discuterons au chapitre 3.

Soulignons une remarque qui peut être utile. La condition de qualification proposée, à savoir que $f'(x^*)$ soit surjective, est peu satisfaisante en ce qu'elle s'applique à x^*, qui est inconnu a priori, ce qui rend le test implicite. On peut s'en dégager, au moins en apparence, grâce à la version équivalente suivante du théorème :

Théorème 1.53 *Si x^* est une solution du problème posé, alors il existe un nombre λ_0 et un vecteur $\lambda \in \mathbb{R}^m$ non tous les deux nuls, tels que*

$$\lambda_0 u'(x^*) + (\lambda, f'(x^*)) = 0\,. \tag{1.39}$$

Demonstration Supposons que $f'(x^*)$ soit surjective. Il suffit de prendre $\lambda_0 = 1$ et (1.39) est établie. Si au contraire $f'(x^*)$ n'est pas surjective, il existe un $\lambda \in \mathbb{R}^m$ non nul tel que $(\lambda, f'(x^*)) = 0$. Il suffit alors de prendre ce λ et $\lambda_0 = 0$ dans (1.39). Le théorème est donc démontré.

On voit que cette version du théorème n'apporte pas grand chose à la version originale. pour s'en convaincre d'avantage, montrons que réciproquement, si la nouvelle version est démontrée elle implique l'ancienne. En effet, supposons donc qu'on ait (1.39) et qu'en outre $f'(x^*)$ est surjective. Alors λ_0 ne peut pas être nul, parce que (1.39) impliquerait alors $(\lambda, f'(x^*)) = 0$ avec $\lambda \neq 0$, contredisant l'hypothèse que $f'(x^*)$ est surjective. Il sufffit alors de diviser λ par λ_0 pour retrouver (1.35).

Conditions du second ordre

Il faut enfin se donner un tout petit peu de mal pour exploiter la condition du second ordre sur $U(x_2) = u(\varphi(x_2), x_2)$, en dérivant deux fois aussi l'égalite $f(\varphi(x_2), x_2) = 0$. Mais on arrive au théorème suivant, qui sera utile dans l'algorithme de la programmation quadratique séquentielle que nous déduirons du théorème de Lagrange :

Théorème 1.54 *Sous la condition de qualification, une condition nécessaire d'optimalité de x^* est que la restriction au noyau de $f'(x^*)$ de la matrice des dérivées secondes du lagrangien soit semi-définie positive. Une condition suffisante locale est que, outre la contrainte $f(x^*) = 0$ et les conditions du premier ordre (1.35), cette même restriction soit positive définie.*

C'est à dire que si λ^* est celui donné par le théorème de Lagrange, on ait en outre

$$\forall y : f'(x^*)y = 0\,, \quad (y, D^2_{xx}\mathcal{L}(x^*, \lambda^*)y) \geq 0\,, \tag{1.40}$$

ou > 0 suivant la condition évoquée.

Notons enfin la remarque suivante, utile dans les algorithmes :

Proposition 1.55 *La propriété "Df surjective et $D^2_{xx}\mathcal{L}$ définie sur le noyau de Df" est équivalente au fait que la matrice*

$$\begin{pmatrix} D^2_{xx}\mathcal{L} & (Df)^t \\ Df & 0 \end{pmatrix}$$

soit inversible.

Chapitre 2

Recherche unidimensionnelle

2.1 Introduction

2.1.1 Objectif

Ce bref chapitre examine une question qui pourrait sembler bien naïve. Elle mérite qu'on s'y attarde un peu parce qu'elle intervient comme technique intermédiaire dans de nombreux algorithmes que nous découvrirons ensuite, c'est la "boucle intérieure" de boucles imbriquées, donc celle qu'il faut soigner le plus.

Soit donc $u(\cdot)$ une fonction d'une seule variable réelle (nous disons "de $\mathbb{R}$ dans $\mathbb{R}$"), dont nous recherchons le minimum sur un intervalle $[a,\, b]$. Nous supposons

- que le minimum t^* recherché est à l'intérieur du segment
- que la fonction u est *unimodale* sur le segment, c'est à dire d'abord décroissante jusqu'à t^*, puis croissante.

Dans la pratique, choisir a et b pourra poser des problèmes, que nous n'examinons pas ici.

Le problème est de déterminer des algorithmes permettant de trouver t^* efficacement, c'est à dire avec une bonne précision mais "pas trop" de calculs.

2.1.2 Pente et dérivée numérique

Suivant les algorithmes proposés, on peut ou non avoir besoin de la "pente" de la fonction, ou plus précisément du signe ou de la valeur de sa dérivée. Dans bien des cas, la dérivée est aussi facile à calculer que la fonction elle-même, et ceci ne pose pas de problème. Mais il y a aussi des problèmes pour lesquels le calcul de la dérivée demande significativement plus de calculs que celui de la fonction. Remarquons que pour une fonction de $\mathbb{R}^n$ dans $\mathbb{R}$, la dérivée consiste en n nombres, soit n fois plus que la fonction.

Il y a aussi des situations pour lesquelles la dérivée en tant que telle n'est pas connue. Typiquement, la fonction u peut n'être donnée que comme un gros programme informatique auquel on fournit t et qui rend $u(t)$. Il est utile de savoir que sont en train d'apparaître des outils informatiques — tels Odyssée— qui prennent en entrée le source d'un programme (FORTRAN pour Odyssée) et produisent en sortie le source (dans le même langage) d'un programme qui calcule les dérivées partielles de la fonction que définit le programme donné. Mais ces outils sont encore du domaine de la recherche, ne marchent que sous certaines conditions sur la façon dont est écrit le programme donné, etc. Supposons, par exemple, que la fonction u soit calculée à l'aide de fonctions intermédiaires tabulées et non disponibles sous forme de combinaison d'opérations et de fonctions "élémentaires". Il

n'y a aucune chance qu'on puisse en calculer formellement la dérivée.

Dans ces cas, on peut avoir recours à la "dérivation numérique", un grand mot pour quelque chose de bien élémentaire. Il s'agit tout simplement d'approximer $u'(t)$ par $(u(t+\delta)-u(t))/\delta$. La difficulté est de choisir δ, assez petit pour que ceci soit une approximation "raisonnable" de la dérivée, mais assez grand pour que la différence $u(t+\delta)-u(t)$ soit calculée de façon significative. On voit que le bon choix de δ dépend de la précision avec laquelle sont faits les calculs. Il faut parfois tatonner pour choisir ce paramètre.

Chaque fois qu'on veut seulement le signe de la dérivée (la réponse à la question "la fonction est-elle croissante ou décroissante en t ?"), pour savoir de quel côté du minimum on se trouve, le "risque" est que ce minimum soit entre t et $t+\delta$, et qu'on ne le remarque pas. Notons à ce propos qu'il est de toutes façons illusoire de chercher le minimum t^* avec une précision meilleure que celle avec laquelle on sait distinguer deux valeurs de t par la valeur qu'elles donnent à $u(t)$. (Sauf si la dérivée est calculable, et avec une meilleure précision que la fonction, cas tout à fait inhabituel.) Mais ceci impose de choisir δ suffisamment petit.

2.2 Méthodes directes

On appelle "méthodes directes" celles qui ne font pas appel au calcul de la dérivée. Nous étendrons cela à celles qui ne demandent que le *signe* de la dérivée.

2.2.1 Dichotomie

La méthode la plus simple, et pas forcément la plus mauvaise, consiste moralement à résoudre $u'=0$ par dichotomie. On arrive ainsi à l'algorithme suivant :

Algorithme Dichotomie

1. $a_0 := a, \qquad b_0 := b$
2. $n = 1$
3. $m := (a_{n-1}+b_{n-1})/2$
4. Évaluer $u'(m)$
5. si $u'(m)<0\,, \quad a_n := m\,, \quad b_n := b_{n-1},$
   si $u'(m)>0\,, \quad a_n := a_{n-1}\,, \quad b_n := m,$
6. Incrémenter n de 1 et retourner au pas 3

On a omis dans l'algorithme ci-dessus le test d'arrêt, qui est un ingrédient *nécessaire* de *tout* algorithme. C'est qu'ici, on sait exactement la précision obtenue au pas n : on peut affirmer que $t^* \in [a_n, b_n]$, donc on a une précision de $(b-a)/2^n$. On peut donc déterminer *a priori* le nombre de pas à effectuer en fonction de la précision souhaitée. On comparera donc n à ce nombre avant de l'incrémenter.

On a choisi, dans la description de l'algorithme ci-dessus, une version compréhensible du point de vue mathématique, et les mathématiques n'aiment pas donner le même nom à des variables différentes. Un informaticien aurait écrit l'algorithme de la façon plus économe suivante :

Algorithme Faire N fois

1. $m := (a+b)/2$
2. Si $u'(m)<0 \quad a := m,$
   si $u'(m)>0 \quad b := m$

3. Recommencer au début

Où N est choisi en fonction de la précision souhaitée. Si cette précision est ε, N croit comme le logarithme (à base 2) de $1/\varepsilon$ (exercice : calculer N). Chaque pas demande un calcul de u', ou, si on doit utiliser des dérivées numériques, deux calculs de u. Le nombre de calculs de u à effectuer croit donc comme $2\log_2(1/\varepsilon) = 2\ln(1/\varepsilon)/\ln 2$.

2.2.2 Suites de Fibonacci

Par une méthode directe, on peut faire mieux que la dichotomie, soit une croissance moins rapide que $2\log_2(1/\varepsilon)$. Le principe de la méthode, dite des *suites de Fibonacci* (on verra pourquoi), est le suivant.

On part du segment $[a_0,\, b_0]$, et on suppose qu'on a calculé $u(a_0)$ et $u(b_0)$. On choisit deux point intérieurs $c_0 < d_0$. On calcule encore $u(c_0)$ et $u(d_0)$. La proposition de base est la suivante :

Proposition 2.1 *Si* $u(c_0) < u(d_0)$, *alors* $t^* \in [a_0,\, d_0]$, *si au contraire* $u(c_0) > u(d_0)$, *alors* $t^* \in [c_0,\, b_0]$.

En effet, en parcourant le segment $[a_0,\, b_0]$, dans le premier cas, la fonction u a nécessairement commencé à croitre avant d_0, et donc $t^* < d_0$, et au contraire dans le deuxième cas, elle a continué à décroitre audelà de c_0, donc $t^* > c_0$.

L'algorithme s'en déduit dans son principe : prendre pour $[a_1,\, b_1]$ le nouveau segment contenant sûrement t^*, $[a_0,\, d_0]$ ou $[c_0,\, b_0]$ suivant le cas, et recommencer. Mais au pas suivant, on connait déjà u en un point intérieur, c_0 dans le premier cas et d_0 dans le deuxième. Donc on n'aura plus qu'à choisir un seul autre point intérieur, et calculer u une seule fois, pour itérer.

La difficulté qui subsiste est dans le choix judicieux des points intérieurs à chaque pas. Le premier choix, naturel, est d'imposer qu'ils soient symétriques par rapport au milieu du segment, ainsi la longueur du segment restant après l'évaluation de u et application de la proposition sera indépendante du résultat du test. En outre, le calcul du deuxième nouveau point intérieur à chaque pas est alors trivial, puisqu'il suffit de prendre le symétrique de celui déjà connu.

Mais cette politique fait dépendre toute la suite des points utilisés du choix des deux premiers (en fait d'un des deux premiers, l'autre étant fixé par symétrie), et cette dépendance peut mener à des blocages. Par exemple, le point intérieur connu au pas n (soit c_{n-1} ou d_{n-1} suivant le cas) pourrait se retrouver au milieu, ou tout près du milieu, du segment, une situation qui empêche d'appliquer notre méthode.

Pour analyser cette question, il faut rentrer un peu dans le détail de cette suite de points.

On supposera qu'on s'est arrangé pour qu'à chaque pas, c_n soit plus grand que le milieu $(a_n + d_n)/2$ de $[a_n,\, d_n]$, et d_n soit plus petit que le milieu $(c_n + b_n)/2$ de $[c_n,\, b_n]$. Ainsi, l'algorithme peut être précisé ainsi :

Algorithme Fibonacci (pas $n+1$)

1. Évaluer u au point intérieur manquant (symétrique de celui déjà connu),
2. si $u(c_n) < u(d_n)$,
   faire $a_{n+1} := a_n, \quad b_{n+1} := d_n, \quad d_{n+1} := c_n, \quad c_{n+1} := a_{n+1} + b_{n+1} - d_{n+1}$,
   si $u(c_n) > u(d_n)$,
   faire $a_{n+1} := c_n, \quad b_{n+1} := b_n, \quad c_{n+1} := d_n, \quad d_{n+1} := a_{n+1} + b_{n+1} - c_{n+1}$.

Appelons $l_0 := b_0 - a_0$ la longueur du segment initial, et de même l_n la longueur du segment $[a_n,\, b_n]$. Du fait de la symétrie, nous avons déjà remarqué que la longueur des segments successifs

ne dépend pas du résultat du test. Examinons donc deux pas consécutifs en supposant que $u(c_{n-1}) < u(d_{n-1})$ et $u(c_n) < u(d_n)$. Ainsi, $[a_n, b_n] = [a_{n-1}, d_{n-1}]$, et $d_n = c_{n-1}$, puis $[a_{n+1}, b_{n+1}] = [a_n, d_n] = [a_{n-1}, c_{n-1}]$. Ainsi, $l_{n+1} = c_{n-1} - a_{n-1}$, tandis que $l_n = d_{n-1} - a_{n-1}$ qui, par symétrie, est aussi $l_n = b_{n-1} - c_{n-1}$. On est donc arrivé à la relation de récurence fondamentale de cette étude :

$$l_{n-1} = l_n + l_{n+1}\,. \tag{2.1}$$

En faisant tourner cette récurence *à l'envers*, c'est à dire en posant $f_k = l_{N-k}$ pour un nombre N suffisamment grand, on voit qu'on arrive à la récurrence

$$f_{k+1} = f_k + f_{k-1}\,, \tag{2.2}$$

qui initialisée en $f_0 = 0$, $f_1 = 1$ donne la suite des nombres de Fibonacci. (exercice : faire un peu de bibliographie pour retrouver qui était Fibonacci, et pourquoi il s'est intéressé à cette suite.)

On peut facilement calculer les premiers termes de la suite de Fibonacci :

$$\begin{array}{rcr} f_0 &=& 0 \\ f_1 &=& 1 \\ f_2 &=& 1 \\ f_3 &=& 2 \\ f_4 &=& 3 \\ f_5 &=& 5 \\ \vdots & & \\ f_{16} &=& 987 \\ f_{21} &=& 10946 \\ f_{26} &=& 121393 \end{array}$$

À l'évidence, la récurrence de Fibonacci génère une suite de nombres entiers positifs qui croit très vite et tends vers l'infini. On peut montrer, et c'est important, qu'elle croit comme ρ_+^k où $\rho_+ = (\sqrt{5}+1)/2$ est connu comme le *nombre d'or*, et est la racine positive de l'équation caractéristique de la récurrence (2.2) :

$$\rho^2 = \rho + 1\,.$$

On aura aussi besoin de $r_+ = 1/\rho_+ = (\sqrt{5}-1)/2$, qui satisfait, lui

$$r^2 = 1 - r\,,$$

dont on voit qu'elle est l'équation caractéristique de la récurence (2.1) écrite $l_{n+1} = l_{n-1} - l_n$, et des racines négatives des mêmes équations $\rho_- = (1-\sqrt{5})/2$ et $r_- = (-1-\sqrt{5})/2$.

La politique "optimale" consiste à avoir au dernier pas une distance δ entre les deux points intérieurs, pour être aussi près que possible de diviser le dernier intervalle par deux. Et le résultat de ce test doit laisser une longueur ε. (Où δ est celui évoqué au titre de la dérivation numérique, et ε la précision souhaitée.)

On en déduit

$$l_N = \varepsilon = l_{N-1}/2 + \delta/2\,,$$

soit $l_{N-1} = 2\varepsilon - \delta$. À partir de là, on peut remonter la suite en utilisant la récurrence (2.1) :

$$\begin{array}{lcccc} l_N & = & \varepsilon & = & f_2\varepsilon \\ l_{N-1} & = & 2\varepsilon - \delta & = & f_3\varepsilon - f_1\delta \\ l_{N-2} & = & 3\varepsilon - \delta & = & f_4\varepsilon - f_2\delta \\ l_{N-3} & = & 5\varepsilon - 2\delta & = & f_5\varepsilon - f_3\delta \\ \vdots & & & & \\ l_0 & = & & & f_{N+2}\varepsilon - f_N\delta \end{array}$$

On doit donc

Algorithme Fibonacci (complet)

1. Déterminer le plus petit entier N tel que $f_{N+2}\varepsilon - f_N\delta > l_0 = b_0 - a_0$,
2. calculer $\varepsilon' = (l_0 + f_N\delta)/f_{N+2}$,
3. choisir $d_0 = a_0 + f_{N+1}\varepsilon' - f_{N-1}\delta$ et $c_0 = b_0 - f_{N+1}\varepsilon' + f_{N-1}\delta$,
4. dérouler l'algorithme de Fibonacci ci-dessus, de $n = 0$ à $N - 1$. (Soit N pas)

Dans cet algorithme, on a évalué u en $N + 3$ points, pour réduire le segment contenant t^* dans un rapport de l'ordre de $1/(\rho_+)^N$. Il est prudent de suivre la suite des nombres de Fibonacci pour placer les points intermédiaires à chaque pas plutôt que de se fier au "symétrique", ce qui évite de laisser s'accumuler de petites erreurs. En effet, cette procédure est *très sensible* à de petites erreurs sur la position des points intérieurs, comme la suite de l'analyse va nous le montrer.

2.2.3 Section dorée

La suite de Fibonacci "à l'envers" engendrée par (2.1) est de la forme

$$l_n = \alpha_+ r_+^n + \alpha_- r_-^n$$

où α_+ et α_- dépendent des deux termes initiaux. Comme r_+ est de module inférieur à 1, le terme en r_+^n tend rapidement vers 0. Par contre, r_- est de module supérieur à 1, et même plus précisément $r_- < -1$, de sorte que le terme en r_-^n diverge rapidement avec des signes alternés. C'est ce qui explique la grande sensibilité de l'algorithme "de Fibonacci" au choix de c_0 et d_0.

La seule façon de pouvoir poursuivre l'algorithme un nombre arbitraire de pas est de s'assurer que $\alpha_- = 0$, c'est à dire de choisir $l_1 = r_+ l_0$. Ainsi, $l_2 = (1 - r_+)l_0 = r_+^2 l_0$, …, $l_n = r_+^n l_0$.

On a ainsi un algorithme bien plus facile à mettre en œuvre, qui consiste à appliquer l'algorithme de Fibonacci ci-dessus, en plaçant à chaque pas d_n à une distance $r_+ l_n$ de a_n, et c_n à une distance $(1 - r_+)l_n = r_+^2 l_n$, où on rappelle que

$$\begin{aligned} r_+ &= \frac{\sqrt{5}-1}{2} \simeq 0,618\,, \\ 1 - r_+ = r_+^2 &= \frac{3-\sqrt{5}}{2} \simeq 0,382\,. \end{aligned}$$

Cet algorithme est à très peu de chose près aussi efficace que le précédent et beaucoup plus simple. On n'a pas besoin de déterminer à l'avance le nombre de pas qu'on va effectuer, il suffit de mettre un test d'arrêt en comparant la longueur du segment $[a_n, b_n]$ restant après le pas n à la précision ε désirée. Aussi le récapitulons-nous à fin de référence.

Algorithme Section dorée

1. Faire $[a_0, b_0] = [a, b]\,, \quad l_0 = b - a\,, \quad c_0 = a + r_+^2 l_0\,, \quad d_0 = a + r_+ l_0.$
2. Évaluer u en a_0, b_0, c_0, d_0.
3. Faire $n := 0$.
4. Calculer $l_{n+1} = r_+(b_n - a_n)$.
5. Si $u(c_n) < u(d_n)$,
   faire $a_{n+1} := a_n, \quad b_{n+1} := d_n, \quad d_{n+1} := c_n, \quad c_{n+1} := a_{n+1} + r_+^2 l_{n+1}$,
   si $u(c_n) > u(d_n)$,
   faire $a_{n+1} := c_n, \quad b_{n+1} := b_n, \quad c_{n+1} := d_n, \quad d_{n+1} := a_{n+1} + r_+ l_{n+1}$.
6. Si $l_{n+1} < \varepsilon$ ou $(\sqrt{5} - 2)l_{n+1} < \delta$, donner $a_{n+1} < t^* < b_{n+1}$ et arrêter,
   si non, évaluer u en celui des points c_{n+1} ou d_{n+1} où elle n'est pas encore connue, incrémenter n de 1 et retourner en 4.

exercice : Pourquoi le deuxième critère dans le test d'arrêt ? (Qui limite la précision qu'il est possible d'obtenir à un peu plus de 4δ. On peut, si vraiment nécessaire, ajouter trois pas de dichotomie.)

2.3 Méthodes indirectes

Comme nous l'avons dit, nous regroupons sous ce titre douteux les méthodes qui reposent de façon plus essentielle sur le calcul de dérivées. Rappelons que dans bien des cas, la recherche unidimensionnelle est effectuée dans un algorithme de minimisation dans $\mathbb{R}^n$ pour trouver le meilleur point dans une *direction de recherche* h choisie par ailleurs. Dans ces cas, notre fonction $u(t)$ de ce chapitre est en fait de la forme $u(x + th)$, et ce qui joue le rôle de $u'(t)$ est la dérivée directionnelle $(\nabla u(x + th), h)$.

2.3.1 "Backtracking"

Sous ce vocable anglosaxon, nous visons une méthode simplissime dont l'objectif n'est pas vraiment de trouver le minimum en t de $u(t)$, mais un point "suffisamment bon". Cette méthode sera recommandée dans l'algorithme de Newton "protégé" de l'optimisation multivariable.

On est au voisinage d'un t donné, et on a calculé $u'(t)$. On cherche un nouveau $t' = t + \theta$. L'idée est de partir avec une estimée trop lointaine $t + \theta_0$, avec $\theta_0 u'(t) < 0$, et de reculer jusqu'à ce que la pente $(u(t + \theta) - u(t))/\theta$ soit suffisamment grande en valeur absolue, comparée à $u'(t)$. On va donc choisir deux nombres positifs $r < 0,5$ et $\rho < 1$ (en pratique on prend ρ vers $0,8$) et de faire

Algorithme

1. choisir θ_0 "siffisamment grand"
2. faire $n := 0$,
3. itérer jusquà ce que $u(t + \theta_n) \leq u(t) + ru'(t)\theta_n : \theta_{n+1} = \rho\theta_n$.

2.3.2 Méthode de Newton

La méthode de dichotomie présentée comme méthode "directe" consiste en fait à résoudre l'équation $u'(x) = 0$ par une méthode de dichotomie. On peut bien sûr résoudre cette même équation par la méthode de Newton. On rappelle que cette méthode itérative consiste à prendre comme prochaine estimée de la solution d'une équation le point qui serait la solution si la fonction à annuler coïncidait avec son approximation au premier ordre.

On verra la méthode de Newton dans un cas un peu plus général au chapitre suivant, nous nous contentons ici d'écrire l'algorithme auquel elle mène pour l'application présente :

Algorithme Newton unidimensionnel

1. Choisir t_0 suffisamment proche de t^*,
2. faire $n := 0$,
3. itérer jusqu'à ce que $|u'(t_n)| \leq \varepsilon$

$$t_{n+1} = t_n - u''(t_n)^{-1} u'(t_n)\,.$$

(Ajouter une clause limitant le nombre total d'itérations permis serait une prudence élémentaire.)

On sait que l'algorithme de Newton converge quadratiquement (on le reverra au chapitre suivant), ce qui est *très rapide*. Sa grande faiblesse est qu'il est très sensible au choix de la condition initiale, et peut facilement être mis en défaut si elle est trop éloignée de la solution recherchée. Aussi, on suggère fréquemment de ne l'utiliser que pour affiner une solution approchée obtenue avec une méthode plus rustique.

On voit aussi sur la formule que cet algorithme aura des problèmes si $u''(t)$ est trop proche de 0 au voisinage de t^*. Il est toujours plus difficile de trouver un minimum très "plat" (avec une très petite dérivée seconde) qu'un minimum bien marqué. Mais cet algorithme, utilisant explicitement la dérivée seconde, peut y être plus sensible qu'un autre. Il peut être prudent de tester le module de cette dérivée seconde (qui devrait rester positive en tout état de cause).

2.3.3 Approximation polynômiale

Approximation parabolique

Dans beaucoup d'applications, notamment celles liées au gradient "à pas optimal" ou "conjugué", on connait u et sa dérivée en a. Si dans le cas du gradient conjugué il est important de trouver le minimum avec une certaine précision, il n'en va pas de même pour le gradient à pas optimal, et pour quelques autres algorithmes où une approximation moyenne de l'optimum suffit à chaque pas. (Relaxation,...) Dans ces cas, on peut utiliser la méthode suivante.

Supposons que nous connaissions u et sa dérivée en a. On choisit b tel que $[a,\, b]$ ait une bonne chance d'encadrer le minimum t^* (mais ce n'est pas critique dans cette méthode), on évalue $u(b)$, on approxime $u(t)$ par la parabole qui aurait même valeur en a et b et même dérivée en a. Et on prend comme estimée de t^* le point $\hat{t}$ où cette parabole atteint son minimum.

Ceci mène à l'estimée suivante :

$$u(t) \simeq u(a) + (t-a)u'(a) + \alpha\frac{(t-a)^2}{2}\,,$$

soit

$$t^* \simeq \hat{t} = a - \frac{u'(a)}{\alpha}\,,$$

où α est estimée par

$$u(b) = u(a) + (b-a)u'(a) + \alpha\frac{(b-a)^2}{2}$$

ce qui conduit finalement au choix

$$\hat{t} = a - \frac{u'(a)(b-a)^2}{2[u(b) - u(a) - (b-a)u'(a)]}\,.$$

On ne traite pas cette méthode comme un algorithme au sens propre du terme, car l'idée n'est pas de l'appliquer itérativement, mais une seule fois dans un algorithme englobant qui requiert d'aller au minimum dans une direction donnée à chacun de ses pas. On évite ainsi des itérations emboitées, au prix d'une approximation parfois médiocre de ce minimum.

Au fur et à mesure que l'algorithme englobant se rapproche du minimum recherché, cette méthode approxime de mieux en mieux le t^* du pas en cours, comme l'indique le résultat ci-dessous.

Théorème 2.2 *Si la fonction u est trois fois continument différentiable, α-convexe sur $[a,\, b]$, et si ce segment contient t^* et a une longueur $b-a$ qui tend vers zéro comme h, $|\hat{t}-t^*|$ tend vers zéro comme h^2. (Donc l'erreur relative $|\hat{t}-t^*|/h$ tend vers zéro comme h.)*

Démonstration Nous donnons une preuve directe de ce résultat, mais il peut aussi se déduire de la preuve plus simple que nous donnons du théorème suivant.

Développons u au voisinage de t^*, en notant $u^* = u(t^*)$ et $u''^* = u''(t^*) > \alpha$:

$$u(t) = u^* + \frac{1}{2}u''^*(t-t^*)^2 + \varepsilon(h^3)$$

où $\varepsilon(z)$ désigne une quantité qui tend vers zéro comme z. On a aussi

$$u'(t) = u''^*(t-t^*) + \varepsilon(h^2)\,.$$

Reportons ces expressions dans la formule pour $\hat{t}$. On trouve facilement

$$u'(a)(b-a)^2 = u''^*(a-t^*)(b-a)^2(1+\varepsilon(h))$$

et

$$u(b) - u(a) - u'(a)(b-a) = \frac{1}{2}u''^*(b-a)^2(1+\varepsilon(h))\,,$$

soit

$$\hat{t} = a - \frac{u''^*(a-t^*)(b-a)^2(1+\varepsilon(h))}{\frac{1}{2}u''^*(b-a)^2(1+\varepsilon(h))} = a - (a-t^*)(1+\varepsilon(h)) = t^* + \varepsilon(h^2)\,.$$

Donc, si cette procédure est utilisée, par exemple, dans un algorithme de gradient, au début, $\hat{t}$ est une approximation grossière du t^* du pas en cours, mais on sait que cela suffit, et au fur et à mesure que l'algorithme progresse, l'erreur relative tend vers zéro, permettant une bonne convergence de l'algorithme global.

Approximations cubiques

En fait, la méthode la plus fréquemment utilisée est celle de l'approximation cubique, où la fonction u est approximée par une cubique, donc. En effet, sa dérivée est alors un polynôme de degré deux, et on a encore une formule explicite pour son minimum $\hat{t}$. Ainsi, au prix de calculs guère plus lourds, on a une estimée meilleure d'un ordre (une erreur relative d'ordre deux), ce qui est excellent.

On prendra donc pour approximation de u

$$u(t) \simeq \alpha t^3 + \beta t^2 + \gamma t + \delta\,,$$

et donc

$$u'(t) \simeq 3\alpha t^2 + 2\beta t + \gamma\,,$$

ce qui mène à l'approximation $t^* \simeq \hat{t}$ avec

$$\hat{t} = \frac{\sqrt{\beta^2 - 3\alpha\gamma} - \beta}{3\alpha} .$$

(On a supposé que $\gamma = u'(0) < 0$ et $\beta = u''(0) > 0$.)

Il reste à calculer α, β et γ. Plusieurs méthodes sont possibles (d'où le pluriel dans le titre de ce sous-paragraphe).

Soit, en calculant $u(b)$, on a facilement aussi la dérivée $u'(b)$, et cela fait assez d'information, soit on considère que calculer une dérivée est plus difficile que la fonction elle même, et on peut alors préférer calculer $u(c)$ en un point intermédiaire $c \in [a,\, b]$.

Dans le premier cas, on peut évaluer les constantes par les formules suivantes :

$$\alpha = \frac{2(u(b) - u(a)) - (b-a)(u'(b) + u'(a))}{(b-a)^3} ,$$

$$2\beta = \frac{u'(b) - u'(a)}{b-a} - 3\alpha(b+a) ,$$

$$\gamma = u'(a) - 3\alpha a^2 - 2\beta a ,$$

voire, pour préserver la symétrie (ce qui peut être numériquement utile) la formule symétrisée pour γ en faisant $1/2$ de cette expression plus la même en b.

Dans le deuxième cas, nous introduisons les quantités

$$\Delta(a,b) = \frac{u(a) - u(b) - u'(a)(a-b)}{(a-b)^2}$$

et de même pour $\Delta(a,c)$, et on peut montrer les formules suivantes :

$$\alpha = \frac{\Delta(a,c) - \Delta(a,b)}{b-c} ,$$

$$\beta = \frac{c\Delta(a,b) - b\Delta(a,c)}{b-c} - 2a\alpha ,$$

$$\gamma = u'(a) - 3\alpha a^2 - 2\beta a .$$

On a le résultat logique :

Théorème 2.3 *Si la fonction u est quatre fois continument différentiable, α-convexe sur $[a,\, b]$, et si ce segment contient t^* et a une longueur $b-a$ qui tend vers zéro comme h, $|\hat{t} - t^*|$ tend vers zéro comme h^3. (Donc l'erreur relative $|\hat{t} - t^*|/h$ tend vers zéro comme h^2.)*

Démonstration Appelons $\hat{u}(t)$ notre approximation polynômiale de u, et $\varepsilon(t) = u(t) - \hat{u}(t)$ l'erreur d'approximation. Le fait essentiel est le suivant :

Proposition 2.4

$$\forall t \in [a,b] , \quad \varepsilon'(t) \to 0 \quad \text{comme } h^3 .$$

En effet, le développement de Taylor à l'ordre 3 en a avec reste exact nous apprend que u peut sécrire

$$u(t) = \bar{u}(t) + \frac{(t-a)^4}{24} u^{(4)}(t') \,, \quad t' \in [a,t]$$

où $\bar{u}$ est un polynôme de degré 3. Si, dans les formules servant à calculer $\hat{u}$ on remplace $u(b)$ et $u(c)$ par $\bar{u}(b)$ et $\bar{u}(c)$, on obtient exactement $\bar{u}$ à la place de $\hat{u}$. Mais on peut vérifier que $\hat{u}(t)$ sécrit aussi (exercice)

$$\hat{u}(t) = u_a(t) + \frac{(t-a)^2(t-c)}{(b-a)^2(b-c)} u(b) + \frac{(t-a)^2(t-b)}{(c-a)^2(c-b)} u(c)$$

où $u_a(t)$ est un polynôme de degré 3 qui ne dépend que de $u(a)$ et $u'(a)$, mais pas de $u(b)$ et $u(c)$. (u_a et sa dérivée coincident avec $u(a)$ et $u'(a)$ en a, et u_a s'annulle en b et en c, ce qui au vu des formules ci-dessus le définit complètement.) Les quantités $u(b)$ et $u(c)$ interviennent (linéairement) dans la formule ci-dessus avec des coefficients uniformément bornés quand $h \to 0$. Donc les coefficients du polynôme $\hat{u}$ approchent ceux de $\bar{u}$ comme $\bar{u}$ approche u en b et en c, c'est à dire comme h^4. Donc leur différence *et sa dérivée* approchent zéro comme h^4. Ainsi $\hat{u}'$ approche $\bar{u}'$ en h^4 uniformément en t. Mais le développement de Taylor de u' en a nous apprend que $\bar{u}'$ approche u' comme h^3 uniformément en t. La proposition est démontrée.

Par définition de $\hat{t}$ on a $\hat{u}'(\hat{t}) = 0$, et donc $u'(\hat{t}) = \varepsilon'(\hat{t})$. En outre, $u'(t^*) = 0$, et donc, comme u est supposée α-convexe, par (1.25) $u'(\hat{t}) \geq \alpha |\hat{t} - t^*|$. Soit bien

$$|\hat{t} - t^*| \leq \frac{1}{\alpha} \varepsilon'(\hat{t})$$

d'où, avec la proposition, le résultat annoncé.

On remarque que cette méthode de preuve, élémentaire si non élégante, s'étend à un ordre quelconque.

Il reste une question ouverte : celui du meilleur choix de c dans la dernière méthode. Y a-t-il un choix de c qui annule le terme d'ordre 3 dans l'erreur $\hat{t} - t^*$? C'est peu probable, parce que ce terme est un polynôme de degré trois en t^*, et fait donc intervenir quatre coefficients. Mais à notre connaissance, cette question n'a jamais été regardée. Il faut dire que les calculs demandent à être faits à la machine, car ils sont *gros* ! (Il faut développer u au moins à l'ordre cinq.)

Chapitre 3

Optimisation dans $\mathbb{R}^n$

3.1 Bonnes fonctions

Nous présentons ci-dessous divers algorithmes de recherche de minimum dont certains, tel le gradient à pas optimal, sont très robustes, i.e. convergent dans bien des situations délicates. Cependant, tant parce qu'on ne peut pas toujours faire beaucoup mieux que par souci de simplicité mathématique, nous ne démontrerons jamais la convergence que pour une classe de fonctions u assez restreinte, que nous appellerons les *bonnes fonctions* (appellation totalement indigène).

Définition 3.1 (Bonnes fonctions) *Nous appellerons* bonnes fonctions *les fonctions $u(\cdot)$ de $\mathbb{R}^n$ dans $\mathbb{R}$ α convexes et de dérivée première β-Lipshitz-continue :*

$$\forall x, y \in \mathbb{R}^n, \quad \|u'(y) - u'(x)\| \leq \beta \|y - x\| \,. \tag{3.1}$$

Si nous définissons la convexité via la positivité de la dérivée seconde, —mais la définition ci-dessus est plus générale— c'est dire que $u(\cdot)$ doit être deux fois continument dérivable, et qu'il doit exister deux réels positifs α et β tels que

$$\forall x \in \mathbb{R}^n, \quad \alpha I \leq D^2 u(x) \leq \beta I \,,$$

ou encore que les valeurs propres de $D^2 u$ sont comprises pour tout x entre α et β.

Rappelons que $u(\cdot)$ étant α-convexe, elle satisfait outre (3.1) les inégalités (1.25), (1.24), (1.18) et (1.16), et son minimum est atteint sur $\mathbb{R}^n$.

3.2 Optimisation non contrainte

3.2.1 Relaxation

Nous commençons par un algorithme "direct", c'est à dire qui ne nécessite pas le calcul des dérivées de u. Il n'est à recommander que si ce calcul est vraiment difficile, et (si possible) seulement en petite dimension.

L'algorithme est excessivement simple (voire simpliste), et consiste à faire des minimisations unidimensionnelles en chacune des variables x_i successivement. Un test d'arrêt raisonnable porte sur la décroissance de u aprés qu'on ait fait cette minimisation sur chacune des coordonnées.

Soit donc x^k une estimée de la solution. Nous introduisons les "pas fractionnaires" $x^{k+i/n}$, que nous noterons $x^{k,i}$ pour simplifier, de la façon suivante : $x^{k,1}$ ne diffère de x^k que par sa première

coordonnée obtenue en minimisant $u(\cdot)$ par rapport à cette première coordonnée toutes les autres étant figées à leur valeur dans x^k. On passe ensuite à $x^{k,2}$ en figeant toutes les coordonnées à leur valeur dans $x^{k,1}$, sauf la deuxième par rapport à laquelle on minimise u, et ainsi de suite. Et on posera $x^{k+1} = x^{k,n}$. Nous décrivons formellement cet algorithme. Nous notons e_i le vecteur de base numéro i de $\mathbb{R}^n$.

Algorithme Relaxation

1. Choisir une estimée initiale x^0 et faire $k := 0$
2. faire $x^{k,0} := x^k$.
3. pour $i = 1 \dots n$, calculer $x^{k,i}$ par
$$u(x^{k,i}) = \min_{\theta} u(x^{k,i-1} + \theta e_i) .$$
4. faire $x^{k+1} := x^{k,n}$
5. si $u(x^k) - u(x^{k+1}) < \varepsilon$, stop, sinon incrémenter k de 1 et retourner en 2.

Théorème 3.1 (Convergence de l'algorithme de relaxation) *Si $u(\cdot)$ est une bonne fonction (au sens de la définition ci-dessus), l'algorithme de relaxation converge vers l'argument x^* du minimum.*

Démonstration On remarquera d'abord que par l'α-convexité, et à cause de (1.16) par exemple, la suite $\{x^k\}$ est bornée.

Par construction, on a $u(x^{k,i}) < u(x^{k,i-1})$, et donc aussi $u(x^{k+1}) < u(x^k)$. Comme par α-convexité, u est bornée inférieurement, $\|u(x^k) - u(x^{k+1})\| \to 0$ quand $k \to \infty$. Plus précisément, la définition de $x^{k,i}$ implique que $u'(x^{k,i})e_i = 0$, de sorte que l'inégalité (1.24) donne

$$u(x^{k,i-1}) - u(x^{k,i}) \geq \frac{\alpha}{2}\|x^{k,i-1} - x^{k,i}\|^2 .$$

En sommant ces inégalités de $i = 1$ à n, il vient

$$u(x^k) - u(x^{k+1}) \geq \frac{\alpha}{2}\|x^k - x^{k+1}\|^2 .$$

Donc, en particulier, $\|x^k - x^{k+1}\|$ tend vers zéro, et donc aussi chacune de ses composantes $\|x^{k,i-1} - x^{k,i}\|$.

La deuxième inégalité d'α-convexité (1.25) donne, en utilisant $\nabla u(x^*) = 0$:

$$\alpha\|x^k - x^*\|^2 \leq (\nabla u(x^k), x^k - x^*) = \sum_{i=1}^{n} u_i'(x^k)(x_i^k - x_i^*) ,$$

où $u_i' = u'(x)e_i$ désigne bien sur la dérivée partielle en x_i. Mais à nouveau, par la définition de $x^{k,i}$, $u_i'(x^{k,i}) = 0$. De sorte que l'inégalité (3.1) donne

$$\|u_i'(x^k)\| \leq \beta\|x^k - x^{k,i}\| .$$

Nous utilisons cette évaluation dans l'inégalité précédente, pour obtenir

$$\alpha\|x^k - x^*\|^2 \leq \beta \sum_{i=1}^{n} \|x^k - x^{k,i}\| \, \|x_i^k - x_i^*\| .$$

Nous savons que les $\|x^k - x^{k,i}\|$ tendent vers zéro et que les $\|x_i^k - x_i^*\|$ sont bornés. Donc $\|x^k - x^*\| \to 0$, ce qu'il fallait démontrer.

Cette preuve ne dit pas que la méthode converge "bien". Elle converge même souvent *très mal*, et il n'est guère réaliste d'en espérer une bonne approximation de x^*. Tout au plus permet-elle d'améliorer parfois significativement la performance $u(x)$ d'une estimée initiale à moindre frais. Les méthodes qui suivent sont presque toujours préférables.

3.2.2 Gradient à pas optimal

L'algorithme

L'algorithme de gradient à pas optimal consiste à se déplacer "dans la direction de plus grande pente", c'est à dire dans la direction opposée au gradient, jusqu'au point "le plus bas" dans cette direction. On a ainsi la description formelle suivante :

Algorithme Gradient à pas optimal

1. Choisir une estimée initiale x^0, faire $k := 0$.
2. Calculer $\nabla u(x^k)$. Si $\|\nabla u(x^k)\| < \varepsilon$ stop. Si non,
3. Calculer $x^{k+1} := x^k - \theta^k \nabla u(x^k)$ où le pas $\theta^k > 0$ est déterminé par

$$u(x^{k+1}) = \min_{\theta} u(x^k - \theta \nabla u(x^k))$$

4. incrémenter k de 1 et retourner en 2

Théorème 3.2 (Convergence de l'algorithme du gradient à pas optimal)
Soit $u(\cdot)$ une bonne fonction. L'algorithme du gradient à pas optimal converge vers l'argument x^* du minimum de u.

Demonstration Nous décomposons cette preuve pour souligner ce que nous apporte chaque hypothèse.

1. La fonction u décroit à chaque pas, mais comme elle est α-convexe, elle est bornée inférieurement. Donc $\|u(x^k) - u(x^{k+1})\| \to 0$.
2. Comme u'' est bornée par βI, et plus précisément grâce à l'inégalité (3.1), la décroissance au pas k est d'au moins $\|\nabla u(x^k)\|^2/2\beta$, comme l'indique le lemme que nous démontrons séparément ci-dessous, utilisé avec $\hat{h} = -g/\|g\|$, et donc $\gamma = 1$. En conséquence, en tenant compte du (1) ci-dessus, $\nabla u(x^k) \to 0$.
3. Par l'α-convexité, et plus précisément par l'inégalité (1.18), on en déduit que $x^k \to x^*$, ce qu'il fallait démontrer.

Remarque 3.1 *On remarque qu'on n'a pas vraiment besoin de l'α-convexité de u. Il suffit (exercice) de supposer que u est strictement convexe et tend vers l'infini à l'infini.*

Il reste à démontrer le lemme. Nous le démontrons dans un cadre un peu plus large, qui nous servira par la suite.

Lemme 3.3 *Soit $g = \nabla u(x)$, et $\hat{h}$ un vecteur de $\mathbb{R}^n$ de norme unité, satisfaisant l'inégalité*

$$(g, \hat{h}) \leq -\gamma \|g\| \tag{3.2}$$

avec $0 < \gamma \leq 1$.

Soit $t^+ \in \mathbb{R}^+$ déterminé par

$$u(x + t^+\hat{h}) = \min_{t \in \mathbb{R}^+} u(x + t\hat{h}) \,,$$

et soit $x^+ = x + t^+\hat{h}$.

Sous l'hypothèse (3.1), on a

$$u(x) - u(x^+) \geq \frac{\gamma^2}{2\beta} \|g\|^2 \tag{3.3}$$

Démonstration Introduisons la fonction $U(t) = u(x + t\hat{h})$. Notons que

$$U'(t) = (\nabla u(x + t\hat{h}), \hat{h})$$

et donc que $U'(0) \leq -\gamma\|g\|$.

Par (3.1), nous avons

$$\|\nabla u(x + t\hat{h}) - g\| \leq \beta t \,,$$

soit, avec l'inégalité de Cauchy-Schwarz,

$$(\nabla u(x + t\hat{h}) - g, \hat{h}) \leq \beta t$$

soit encore

$$U'(t) = (\nabla u(x + t\hat{h}), \hat{h}) \leq (g, \hat{h}) + \beta t \leq -\gamma\|g\| + \beta t \,.$$

On utilise cette minoration pour évaluer $U(\tau)$:

$$U(\tau) \leq U(0) + \int_0^\tau (\beta t - \gamma\|g\|)\, dt = u(x) + \beta\frac{\tau^2}{2} - \gamma\|g\|\tau \,.$$

Enfin, on utilise cette minoration en $\tau = \frac{\gamma}{\beta}\|g\|$ pour obtenir

$$u(x^+) = \min_\tau U(\tau) \leq U(\frac{\gamma}{\beta}\|g\|) \leq u(x) - \frac{\gamma^2}{2\beta}\|g\|^2 \,,$$

ce qu'il fallait démontrer.

Cet algorithme est très robuste, et beaucoup plus efficace que l'algorithme de relaxation. On démontre sa convergence pour les "bonnes fonctions", mais c'est un "bon algorithme" au sens où "une bonne théorie est une théorie qui continue à donner de bons résultats quand on l'utilise en dehors des hypothèses sous lesquelles elle a été établie"[1]. Naturellement, en l'absence de convexité, il converge vers le minimum local dans le "bassin d'attraction" duquel on est parti. Il convient donc éventuellement de refaire fonctionner l'algorithme avec divers conditions initiales.

Par contre, on ne doit pas attendre une bonne convergence *près de l'optimum*. C'est à dire que l'algorithme est efficace pour faire décroitre rapidement la fonction, mais pas pour avoir une bonne approximation de x^*. Au point que Claude Lemaréchal a pu écrire dans un autre cours "on devrait **interdire** cette méthode". C'est un peu... totalitaire !. Mais pour approximer finement x^*, il vaut mieux finir avec une méthode du second ordre comme celle de l'algorithme que nous présenterons ensuite.

[1] Holt Ashley

Préconditionnement

On remarque que la preuve de convergence demeure si la direction de descente choisie "fait un angle aigu" avec $-\nabla u$, comme le lemme le montre. Cette remarque a de nombreuses applications, qui tournent autour de ce qu'on appelle le "préconditionnement".

Supposons qu'on fasse un changement de variable $x = A\xi$, avec une matrice de changement de variable A non singulière. On est conduit à considérer la fonction $v(\xi) = u(A\xi)$. On peut montrer (exercice) que c'est une "bonne fonction", et qu'on peut donc lui appliquer l'algorithme du gradient. Cela donne-t-il la même suite de points que l'algorithme initial ? La réponse est non comme nous allons le voir.

Nous avons $v'(\xi) = u'(A\xi)A$, soit en transposant $\nabla v(\xi) = A^t \nabla u(A\xi)$ et donc, $v(\xi - \theta\nabla v) = u(A\xi - \theta AA^t\nabla u)$. On a donc comme direction de descente $-AA^t g$. Si A est régulière, AA^t est positive définie, et il existe donc $\gamma > 0$ tel que $(g, AA^t g) \geq \gamma\|g\|^2$. ($\gamma$ est le carré de la plus petite *valeur singulière* de A.) Donc, de la façon dont a été faite la preuve ci-dessus, on déduit immédiatement que l'algorithme converge encore. Mieux ?, moins bien ? Cela dépend évidemment du choix de A, ou de la matrice symétrique AA^t, qui est appelée matrice de *préconditionnement*.

Les algorithmes de gradient conjugué, par exemple, peuvent être vus come des algorithmes de gradient à préconditionnement soigné, et la méthode de Newton protégée ci-dessous comme un préconditionnement "optimal".

Plus simplement, on recommande en général d'utiliser au moins un préconditionnement diagonal de la forme $\xi_i = x_i/\bar{x}_i$ où les $\bar{x}_i$ jouent le rôle de facteur d'échelle, rendant en quelque sorte les ξ_i sans dimension, et doivent être choisis de façon que la sensibilité de $v(\xi)$ aux différents ξ_i soit à peu près la même. Ce qui est obtenu en prenant des $\bar{x}_i$ inversement proportionnels aux (ordre de grandeur des) $\nabla_i u$.

3.2.3 Méthode de Newton

La méthode "pure"

Nous présentons maintenant une méthode "du second ordre", en ce qu'elle utilise les dérivées secondes de u, mais aussi en ce qu'elle converge (plus vite que) quadratiquement. C'est dans cette famille de méthodes qu'il faut chercher si on veut approximer finement l'argument du minimum x^*.

Le principe est simple, il consiste à résoudre l'équation $u'(x) = 0$ par la méthode de Newton. Nous la rappelons d'abord pour la solution d'une équation $g(x) = 0$ où $g(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$.

La méthode de Newton consiste à approximer g au premier ordre autour du dernier point connu, et à prendre comme prochaine estimée de la solution x^* de $g(x) = 0$, le point où cette approximation linéaire s'annule. Ceci mène à

$$g(x) \simeq g(x^k) + g'(x^k)(x - x^k)\,,$$

soit

$$x^{k+1} = x^k - [g'(x^k)]^{-1} g(x^k) \tag{3.4}$$

Cette méthode converge quadratiquement comme le montre le résultat suivant :

Théorème 3.4 (Convergence de la méthode de Newton) *Si la fonction g est deux fois continument différentiable au voisinage de x^*, avec une dérivée première inversible en x^*, il existe un voisinage de x^* dans lequel la méthode de Newton converge quadratiquement.*

Démonstration Par continuité de g', il existe un voisinage de $\mathcal{V}$ de x^* dans lequel la matrice $g'(x)$ est inversible, et plus précisément a une plus petite valeur singulière bornée inférieurement par un nombre positif α, de sorte que $[g'(x)]^{-1}$ existe et a une norme bornée supérieurement par $1/\alpha$. De même, dans ce voisinage, $g''(x)$ existe et a une norme bornée par un nombre positif γ.

Écrivons l'itération de Newton comme

$$x^{k+1} - x^* = [g'(x^k)]^{-1}[g'(x^k)(x^k - x^*) - g(x^k)]$$

Par le développement limité (1.22), et en se souvenant que par définition $g(x^*) = 0$, le dernier crochet ci-dessus peut se ré-écrire comme ci-dessous, coordonnée par coordonnée :

$$g_i'(x^k)(x^k - x^*) - g_i(x^k) = \frac{1}{2}\left((x^k - x^*),\, D^2 g_i(x^k + \theta(x^* - x^k))(x^k - x^*)\right) ,$$

de sorte que ce crochet est borné en norme par

$$\|g'(x^k)(x^k - x^*) - g(x^k)\| \leq \frac{\gamma}{2}\|x^k - x^*\|^2 .$$

Donc, en utilisant la borne sur $\|(g')^{-1}\|$,

$$\|x^{k+1} - x^*\| \leq \frac{\gamma}{2\alpha}\|x^k - x^*\|^2 .$$

Il reste à s'assurer que si on part suffisamment prés de x^*, la suite engendrée ne sort pas de $\mathcal{V}$, ce qui se fait en bornant la somme des $\|x^k - x^*\|$ pour $\|x^0 - x^*\|$ suffisamment petit. Ainsi, le théorème est démontré.

On voit que si $g(x) = \nabla u(x)$, nous avons donné un algorithme de recherche de minimum dans un ouvert, à savoir

$$x^{k+1} = x^k - \left[D^2 u(x^k)\right]^{-1} \nabla u(x^k) , \tag{3.5}$$

et montré sa convergence quadratique si u est trois fois continument différentiable. La grande faiblesse de cet algorithme *très rapide* est sa *très grande sensibilité aux conditions intiales*. C'est pourquoi on conseille souvent de commencer la recherche du minimum avec une méthode plus robuste comme celle du gradient, et de finir avec la méthode de Newton.

Remarque 3.2 *Une remarque importante est que dans cette version "pure", on n'a pas besoin d'inverser $D^2u(x^k)$ en dépit de ce que semble indiquer la formule (3.5). En effet, il suffit de résoudre le système linéaire*

$$D^2 u(x^k)(x^{k+1} - x^k) = -\nabla u(x^k) , \tag{3.6}$$

ce qui peut être significativement moins long. Cela n'évite pas le calcul de la matrice des dérivées secondes $D^2u(x^k)$ à chaque pas.

La méthode de Newton "protégée"

Une méthode recommandée consiste à considérer $h = -[g'(x^k)]^{-1}g(x^k)$ comme une *direction de descente*, et, comme précédemment effectuer une recherche unidimensionnelle de minimum dans cette direction, sachant que l'optimum devrait se trouver près de 1. Le caractère positif défini de $g'(x) = D^2u(x)$, si u est α-convexe, garantit la convergence de cet algorithme en vertu de la preuve de convergence de l'algorithme de gradient. En pratique, la direction de recherche est si bonne qu'il suffit de faire du "backtracking" (cf. le paragraphe 2.3.1) depuis le "pas de Newton" 1.

Il n'est pas très sûr que le poids du calcul de $[D^2u(x^k)]^{-1}$ en vaille la peine si on est vraiment trop loin de l'optimum. Divers aménagement de la méthode de Newton visent à alléger cette étape. Avant de les évoquer, notons que, toujours si la fonction u est convexe, la matrice $\nabla g = D^2u$ à inverser est positive définie, de sorte qu'on dispose pour cette inversion de la très efficace méthode de Cholesky.

Les méthodes de Newton modifiées, quasi-Newton

Le coût principal en calcul dans la méthode de Newton est dans l'inversion de $D^2u(x^k)$ à chaque pas. Remarquons que la preuve de convergence demeure inchangée si on remplace $D^2u(x^k)$ par $D^2u(x^*)$, qui lui, est constant, et ne demanderait donc qu'une inversion unique. Bien sûr, on ne connait pas x^*, et donc pas non plus $D^2u(x^*)$ (sauf si $D^2u(x)$ est constante, mais alors u est une simple fonction quadratique, et l'algorithme de Newton converge en un pas !) Par contre, la preuve demeure encore si on remplace $D^2u(x^k)$ par une matrice H_k telle que

$$\|H_k - D^2u(x^*)\| \leq \eta\|x^k - x^*\| \,.$$

On aura en effet,

$$x^{k+1} - x^* = H_k^{-1}[H_k(x^k - x^*) - g(x^k)] \,,$$

et le crochet s'écrit maintenant

$$H_k(x^k - x^*) - g(x^k) = D^2u(x^*)(x^k - x^*) - g(x^k) + (H_k - D^2u(x^*))(x^k - x^*)$$

dont le module est encore majoré par un terme quadratique en $\|x^k - x^*\|$.

On va donc chercher à avoir, à moindre prix, une suite H_k tendant vers $D^2u(x^*)$ avec x^k. Une idée simplissime, mais assez efficace si elle est utilisée avec doigté, consiste à ne remettre à jour $D^2u(x)$, et donc $D^2u(x)^{-1}$, que de temps en temps. Typiquement tout les deux à cinq pas.

Des méthodes plus sophistiquées existent, sous le nom de méthodes de "quasi Newton", qui s'apparentent au gradient conjugué, et cherchent une formule itérative pour approximer H_k^{-1}. Nous n'en parlerons pas plus ici.

3.3 Optimisation sous contraintes inégalité

3.3.1 Position du problème

La plupart des problèmes d'optimisation, notamment en théorie de la décision, (contrôle, économie théorique, recherche opérationnelle, etc.) se présentent avec des contraintes sur les variables de décision x. D'une manière abstraite, ces contraintes se traduisent par l'existence d'un ensemble $C \subset \mathbb{R}^n$ de *variables admissibles*. On cherche donc $x^* \in C$ tel que

$$\forall x \in C \quad u(x) \geq u(x^*) \,.$$

L'algorithme du gradient projeté ci-dessous considère le problème sous cette forme, et utilisera la projection sur C en le supposant convexe fermé. Ceci suppose que cette projection soit facile à faire, ce qui est rarement le cas.

En pratique, le cas le plus courant est celui où l'ensemble des variables admissibles est défini indirectement par des contraintes de la forme

$$C = \{x \in \mathbb{R}^n \mid f_i(x) \leq 0 \,, \quad i = 1, 2, \ldots, p\} \,,$$

qu'on écrira aussi $f(x) \leq 0$. Les méthodes intéressantes sont alors celles qui sont explicites en fonction de f.

La théorie de la dualité (multiplicateurs de Lagrange, de Kuhn et Tucker, etc) est à la base de plusieurs algorithmes visant à répondre à ce type de problème. Nous ne présenterons, dans cette famille, que l'algorithme d'Uzawa, réputé le plus robuste.

Enfin, par souci de présenter les méthodes les plus employées, nous donnerons des méthodes de pénalisation, qui échangent une plus grande simplicité théorique contre une efficacité douteuse.

3.3.2 Gradient projeté

Projection sur un convexe simple

On a rappelé dans le chapitre premier l'existence de la projection $P_C(x)$ d'un vecteur x sur un convexe fermé C. Cette projection est une opération simple dans un petit nombre de cas. Typiquement trois cas :

- C est un pavé, de la forme $a_i \leq x_i \leq b_i$, où les a_i et b_i sont donnés. Alors la projection se fait coordonnée par coordonnée et consiste simplement à "ramener x_i dans $[a_i,\, b_i]$". On peut écrire cela comme
$$(P_C(x))_i = \max\{a_i,\, \min\{x_i,\, b_i\}\}\,.$$
- C est une boule, de la forme $\|x - \xi\| \leq \rho$ où le vecteur ξ de $\mathbb{R}^n$ et le réel positif ρ sont donnés. Alors la projection consiste à ramener x dans la boule le long du rayon, soit
$$P_C(x) = \xi + \min\{1, \frac{\rho}{\|x - \xi\|}\}(x - \xi)\,.$$
- C est un demi-espace, de la forme $(p, x) \leq a$, où le vecteur p de $\mathbb{R}^n$ et le réel a sont donnés. La projection consiste à ramener x dans le demi espace parallèlement à p :
$$P_C(x) = x - \max\left\{0, \frac{(p,x) - a}{(p,p)}\right\} p\,.$$

Exercice 3.1 *Vérifier les formules ci-dessus.*

L'algorithme

L'algorithme du gradient projeté décrit ci-dessous n'existe que dans cette version à pas fixe. A notre connaissance, il n'y a pas de version "à pas optimal".

Naturellement, si le convexe des contraintes sur lequel on projette est tout $\mathbb{R}^n$, la projection est l'identité, et donc la preuve ci-dessous montre la convergence de l'algorithme de gradient à pas fixe sans projection, pour le problème sans contraintes. (Pourvu, toujours, qu'on ait choisi le pas convenablement.) En fait, dans le cas sans contrainte, l'algorithme à pas fixe n'est *vraiment pas* à recommander.

L'algorithme de gradient projeté est fondé sur la remarque suivante. L'inégalité d'Euler pour l'optimisation dans un convexe (1.23) nous dit que si x^* fournit le minimum de u sur C, alors
$$\forall t > 0 \quad P_C(x^* - t\nabla u(x^*)) = x^*\,. \tag{3.7}$$

En effet, soit x^* est intérieur à C, et alors $\nabla u(x^*) = 0$, ce qu'implique (3.7) car pour un point intérieur, et si $\nabla u(x^*) \neq 0$, il existe t suffisamment petit pour que $x^* - t\nabla u(x^*) \in C$, de sorte qu'il

serait sa propre projection. Soit x^* est un point frontière, et (3.7) est équivalent à l'affirmation que $-\nabla u(x^*)$ est une normale extérieure à C, ce que dit (1.23).

L'algorithme s'écrit comme une méthode de recherche du point fixe dans (3.7) :

Algorithme Gradient projeté

1. Choisir une éstimée initiale x^0, un pas $t > 0$, faire $k := 0$,
2. calculer $x^{k+1} = P_C(x^k - \theta\nabla u(x^k))$,
3. si $\|x_{k+1} - x_k\| \leq \varepsilon$ stop, si non, incrémenter k de 1 et retourner en (2)

On a le résultat de convergence suivant :

Théorème 3.5 (Convergence de l'algorithme du gradient projeté) *Si $u(\cdot)$ est une bonne fonction, et si $0 < \theta < 2/\beta$, l'algorithme de gradient projeté converge vers le minimum x^* de u sur C.*

Remarque 3.3 *Le test d'arrêt ne peut pas porter sur le module du gradient de u, dont on ne sait pas a priori s'il sera grand ou petit. Le test proposé ici vérifie donc qu'il soit "bien orthogonal" au bord de C si x^k est sur ce bord, et petit si-non. En fait, pour l'utilisation avec une fonction u dont la convexité est douteuse, il vaut mieux faire porter ce test sur la différence $\|x^{k+1} - x^{k-4}\|$ par exemple, c'est à dire prendre en considération l'évolution de l'algorithme sur* plusieurs *pas.*

Démonstration Puisque nous avons reconnu dans l'algorithme une itération de point fixe (aussi dite de Picard), montrons que la fonction $x \mapsto P_C(x - \theta\nabla u(x))$ est une contraction pour θ choisi comme dans le théorème. On sait (théorème 1.12) que la projection sur C est une contraction au sens large. Il suffit donc de montrer que $\varphi_\theta(x) := x - \theta\nabla u(x)$ est une contraction stricte. On a $\nabla\varphi_\theta(x) = I - \theta D^2u(x)$. Cette matrice est symétrique. Sa norme est donc son rayon spectral, le module de sa valeur propre de plus grand module. Or ses valeurs propres sont de la forme $1 - \theta\lambda(D^2u(x))$, où les $\lambda(D^2u(x))$ sont les valeurs propres de $D^2u(x)$. Par hypothèses, celles-ci sont comprises entre α et β. Il suffit donc de choisir θ de façon que $|1 - \theta\alpha| < 1$ et $|1 - \theta\beta| < 1$, soit $\theta > 0$ et $\theta < 2/\beta$ respectivement. (Cette dernière condition assurant aussi $\theta < 1/\alpha$.) Ce qui établit la convergence de l'algorithme sous la condition annoncée.

On peut se demander quel est le θ "optimal", ou au moins celui pour lequel le module de Lipshitz de φ_θ est minimal. On voit assez facilement qu'il est tel que $1 - \theta\alpha = \theta\beta - 1$, soit $\theta = 2/(\alpha + \beta)$. Et alors, le module de Lipshitz de φ_θ est $1 - 2\alpha/(\alpha + \beta)$.

On voit que ce nombre est d'autant plus proche de 1 que α est petit. C'est une constante des problèmes d'optimisation que trouver le minimum est d'autant plus difficile que le module d'alpha convexité est petit.

En fait, ce théorème souligne surtout la principale difficulté liée à l'utilisation de cet algorithme. C'est celle du choix du pas θ. En effet, en général, α et β ne sont pas connus —bien heureux s'ils existent—, et il faut donc des heuristiques d'adaptation du pas θ.

Remarquons d'abord qu'à en croire le calcul précédent, un pas trop petit peut ralentir considérablement la convergence, pas l'empêcher comme peut le faire un pas trop grand. Il faut quand même trouver un moyen d'apprécier le pas qu'on peut se permettre. Une règle qu'on peut proposer est la suivante. Comparer la décroissance de u obtenue, $u(x^k) - u(x^{k+1})$, à son estimation au premier ordre $u'(x^k)(x^k - x^{k+1}) = \theta\|\nabla u(x^k)\|^2$ (qui doit être plus grande). Si ces deux quantités coincident "très bien", disons à 10% près, on peut sans doute doubler le pas. Si elles coincident mal, disons plus mal que dans un rapport deux, il faut sans doute diviser le pas par deux.

Cet algorithme n'est sans doute pas très performant par rapport à ceux que nous avons vus jusqu'ici. Il faut bien comprendre que la minimisation sous contrainte est un problème plus difficile que la minimisation sans contrainte, et qu'on ne peut espérer trouver des algorithmes aussi efficaces.

3.3.3 Algorithme d'Uzawa

L'algorithme d'Uzawa exploite le point selle du théorème de Kuhn et Tucker. Il va donc chercher à minimiser le lagrangien par rapport à x et à le maximiser par rapport à la variable duale, disons p. Plus précisément, il consiste à remettre à jour l'estimée courante de la variable duale par un pas de gradient —et le gradient du lagrangien par rapport à p est particulièrement simple—, puis à p fixé, à aller jusqu'au minimum en x. Ce sera donc un "méta algorithme", puisque nous ne dirons pas comment effectuer la minimisation en x. Observons pourtant, —et c'est tout ce que la dualité fait pour nous—, que dans cette minimisation, les contraintes $f(x) \leq 0$ ont disparu.

Comme dans le théorème 1.50, nous permettons ici à certaines contraintes de rester "abstraites" sous la forme $x \in C$ tandis que d'autres sont rendues "concretes" sous la forme $f(x) \leq 0$. La minimisation du lagrangien est alors à effectuer *dans* C. Si C est tout $\mathbb{R}^n$, on a alors affaire à un problème de minimisation *sans contrainte*. Donc un algorithme de gradient à pas optimal, par exemple, est possible. La dualité a ainsi ramené un problème de minimisation sous contraintes à une suite de problèmes de minimisation sans contrainte.

On note P_+ la projection sur le cône positif de $\mathbb{R}^n$, qui consiste juste à ramener à zéro toute composante négative du vecteur à projeter.

Algorithme Uzawa

1. Choisir une estimée initiale $p^0 \geq 0$ Faire $k := 0$.
2. Calculer x^k par
$$u(x^k) + (p^k, f(x^k)) = \min_{x \in C} [u(x) + (p^k, f(x))] .$$
3. faire
$$p^{k+1} = P_+[p^k + \rho f(x^k)] .$$
Si $\|p^{k+1} - p^k\| \leq \varepsilon$, stop. Sinon, retourner en 2

Cet algorithme est en fait un algorithme de gradient (en p) projeté (sur le cône positif). On démontre sa convergence d'une façon analogue à la preuve de convergence de cet algorithme. (Cf théorème 3.5)

Théorème 3.6 (Convergence de l'algorithme d'Uzawa) *Si $u(\cdot)$ est une bonne fonction, et $f(\cdot)$ est Lipshitz-continue de coefficient γ, pour $\rho < 2\alpha/\gamma^2$, la suite $\{x^k\}$ engendrée par l'algorithme d'Uzawa converge vers l'optimum x^*.*

Démonstration Remarquons d'abord que la condition des écarts complémentaires entraine que pour tout $\rho > 0$,
$$p^* = P_+[p^* + \rho f(x^*)] .$$
Ainsi, par le théorème 1.12, on a
$$\|p^{k+1} - p^*\| \leq \|p^k - p^* + \rho(f(x^k) - f(x^*))\| .$$
En élevant au carré, il vient
$$\|p^{k+1} - p^*\|^2 \leq \|p^k - p^*\|^2 + 2\rho(p^k - p^*, f(x^k) - f(x^*)) + \rho^2 \|f(x^k) - f(x^*)\|^2 \qquad (3.8)$$

Nous allons majorer le double produit (le terme central du deuxième membre). Remarquons que, grâce au fait que les p_i sont positifs, $u + (p, f)$ est encore convexe, et même α-convexe, en x. Par (1.23), la minoration (1.16) reste valide pour la minimisation dans un convexe fermé. On a donc

$$\begin{aligned} u(x^k) + (p^k, f(x^k)) &\leq u(x^*) + (p^k, f(x^*)) - \frac{\alpha}{2}\|x^k - x^*\|^2 , \\ u(x^*) + (p^*, f(x^*)) &\leq u(x^k) + (p^*, f(x^k)) - \frac{\alpha}{2}\|x^k - x^*\|^2 . \end{aligned}$$

Sommons membre à membre et faisons passer ce qu'il faut à gauche, pour obtenir

$$(p^k - p^*, f(x^k) - f(x^*)) \leq -\alpha\|x^k - x^*\|^2 .$$

Ensuite, l'hypothèse que f est Lipshitz de coefficient γ donne

$$\|f(x^k) - f(x^*)\| \leq \gamma\|x^k - x^*\| .$$

En reportant ces deux majorations dans (3.8), il vient

$$\|p^{k+1} - p^*\|^2 \leq \|p^k - p^*\|^2 + (\rho^2\gamma^2 - 2\rho\alpha)\|x^k - x^*\|^2 .$$

Si on a choisi $\rho < 2\alpha/\gamma^2$, il existe $\delta > 0$ tel que $\rho^2\gamma^2 - 2\rho\alpha < -\delta$, d'où

$$\|p^{k+1} - p^*\|^2 \leq \|p^k - p^*\|^2 - \delta\|x^k - x^*\|^2$$

qu'on peut ré-écrire

$$\delta\|x^k - x^*\|^2 \leq \|p^k - p^*\|^2 - \|p^{k+1} - p^*\|^2 .$$

Donc, la suite $\|p^k - p^*\|^2$ est décroissante. Comme elle est composée d'éléments positifs, elle converge, donc la différence du deuxième membre ci-dessus tend vers zéro. Donc $\|x^k - x^*\|$ tends vers zéro, ce qu'il fallait démontrer.

Terminons en évoquant l'interprétation "économique" du théorème de Kuhn et Tucker, et donc de l'algorithme d'Uzawa. Si les containtes $f_i(x) \leq 0$ représentent des ressources "rares" dont on ne peut utiliser que la quantité disponible, et les p_i^* associés en sont les prix unitaires à l'équilibre, on voit que le lagrangien est un coût économique total prenant en compte le "prix" des denrées rares utilisées, et que l'algorithme consiste tout simplement à diminuer le prix des ressources qu'on n'utilise pas jusqu'à saturation, et à augmenter le prix de celles pour lesquelles la demande dépasse la disponibilité. Rien que de très naturel.

Dualisation partielle Une remarque importante doit être faite à ce stade. On a énoncé le théorème de Kuhn et Tucker pour un problème de la forme

$$\forall x \in K \quad u(x) \geq u(x^*) .$$

avec

$$K = C \cap \{x \in \mathbb{R}^n \mid f(x) \leq 0\} ,$$

et on a "dualisé" —c'est à dire introduit dans le lagrangien— les seules contraintes "concretes" $f(x) \leq 0$. Les autres sont restées "abstraites" et prises en compte par la condition $x \in C$ dans les inéquations du point selle (1.32) et l'algorithme d'Uzawa.

Si C n'est en effet pas tout $\mathbb{R}^n$, i.e. une partie des contraintes est restée non dualisée, l'étape de calcul de x^k dans l'algorithme est un problème de minimisation sous ces contraintes. Cette minimisation peut être faite à l'aide d'un autre algorithme qu'Uzawa et la dualité, menant à une "hybridation

d'algorithmes". Si C est un convexe simple, cette minimisation peut être effectuée par un algorithme de gradient projeté par exemple.

Bien sûr, le choix des contraintes à exprimer d'une façon ou de l'autre (à dualiser ou à ne pas dualiser) est laissé à l'utilisateur, et il n'est efficace d'utiliser une dualisation partielle qu'en ne laissant non dualisées que des contraintes simples. Typiquement, si on a un nombre important de contraintes de borne $a_i \leq x_i \leq b_i$, qui impliqueraient deux fois plus de multiplicateurs. On peut alors préférer renoncer à les dualiser et les prendre en compte directement dans la minimsation du lagrangien par projection.

3.3.4 Pénalisation

Les méthodes présentées ici essayent de ramener le problème contraint à un problème non contraint de façon plus "naïve", mais qui peut marcher. En particulier, on les utilise de façon non itérative, contrairement à Uzawa. On ne résoudra donc qu'un, ou quelques, problème(s) de minimisation sans contrainte. L'idée est la suivante : on va "faire payer" au critère le fait pour x de sortir de C. Et si ce "prix" est très élevé, l'optimum se trouvera dans C.

Pénalisation extérieure quadratique

Supposons donc que l'ensemble des x admissibles est donné par

$$f_i(x) \leq 0, \quad i = 1, \ldots, p$$

que nous écrivons aussi $f(x) \leq 0$, où $f : \mathbb{R}^n \to \mathbb{R}^p$. Introduisons en outre la fonction $f^+(x)$ appelée *partie positive* de f, définie par

$$f_i(x) = \max\{0\,,\, f_i(x)\}\,, \quad i = 1, \ldots, p\,.$$

Notons la proposition :

Proposition 3.7 *Si f est dérivable, $\|f^+\|^2$ l'est aussi, et on a*

$$\frac{d\|f^+(x)\|^2}{dx} = 2(f^+(x))^t f'(x)$$

Démonstration Notons d'abord que la proposition n'est pas (entièrement) évidente, car f^+, elle, n'est en général pas dérivable aux points où $f(x) = 0$, donc notamment en x^*. Prenons le cas d'une fonction f *scalaire*. Il suffira ensuite d'appliquer le résultat à chaque composante de f.

Remarquons d'abord que si $f(x) > 0$ ou $f(x) < 0$, par continuité l'inégalité reste vraie dans un voisinage de x. Dans le premier cas $f^+ = f$ et dans le deuxième $f^+ = 0$ dans ce voisinage. La formule $[(f^+)^2]' = 2(f^+)f'$ est alors évidemment vérifiée.

Soit donc x un point où $f(x) = 0$ (c'est à dire un point frontière de C). Soit e_i un vecteur de base de $\mathbb{R}^n$. On a donc dans le "quotient différentiel"

$$\frac{1}{t}[(f^+(x + te_i))^2 - (f^+(x))^2] = \frac{1}{t}(f^+(x + te_i))^2$$

et

$$0 \leq \left|\frac{1}{t}(f^+(x + te_i))^2\right| \leq \left|\frac{1}{t}(f(x + te_i))^2\right|\,,$$

la dernière inégalité parce que dans tous les cas, $|f^+(x)| \leq |f(x)|$. Dans les inégalités ci-dessus, f étant dérivable, le terme de droite tends vers $2|f(x)f'(x)| = 0$, et donc le terme central a une limite, qui est zéro. On a donc démontré que $(f^+)^2$ a en x une dérivée partielle en x_i, qui est nulle. Ceci achève de démontrer la proposition.

Nous considérerons le *critère augmenté*

$$u_\varepsilon(x) = u(x) + \frac{1}{\varepsilon}\|f^+(x)\|^2 .$$

La méthode de pénalisation consiste à utiliser la solution x_ε du problème *sans contrainte* :

$$u_\varepsilon(x_\varepsilon) = \min_{x\in\mathbb{R}^n} u_\varepsilon(x)$$

comme approximée de la solution x^* du problème contraint. Cette méthode est justifiée par le résultat suivant :

Théorème 3.8 *Si $u(\cdot)$ est une bonne fonction, et $f(\cdot)$ est continue, $x_\varepsilon \to x^*$ quand $\varepsilon \to 0$.*

Démonstration On a manifestement

$$u_\varepsilon(x_\varepsilon) \leq u_\varepsilon(x^*) = u(x^*) ,$$

la dernière égalité parce que par définition, $f^+(x^*) = 0$. Si u est α-convexe, elle est bornée inférieurement. Donc l'inégalité

$$u(x_\varepsilon) + \frac{1}{\varepsilon} f^+(x_\varepsilon)^2 \leq u(x^*)$$

implique que $f^+(x_\varepsilon) \to 0$ quand $\varepsilon \to 0$. A fortiori, elle implique aussi que $u(x_\varepsilon)$ reste bornée, donc, toujours avec l'α-convexité, que x_ε reste borné. Ainsi, pour toute suite décroissante de ε_k tendant vers zéro, les x_{ε_k} ont des points d'accumulation. Soit $\bar{x}$ un tel point et ε' une suite telle que les $x_{\varepsilon'} \to \bar{x}$. Par continuité de f^+, $f^+(\bar{x}) = 0$, et $\bar{x}$ est donc admissible.

On a aussi

$$u(x_\varepsilon) \leq u_\varepsilon(x_\varepsilon) \leq u(x^*) .$$

Donc par passage à la limite, (u est continue), $u(\bar{x}) \leq u(x^*)$. Comme $\bar{x}$ est admissible, il en découle que $u(\bar{x}) = u(x^*)$ et $\bar{x}$ est optimal. Par l'α-convexité de u, l'optimum est unique. Donc c'est toute la suite qui converge vers x^*.

On démontre aussi que, si la matrice $f'(x^*)$ est injective, ce qui est une hypothèse naturelle, le produit $(1/\varepsilon)f^+(x_\varepsilon)$ tend vers un vecteur λ de $\mathbb{R}^p$, de sorte qu'à l'optimum on a $u'(x^*) + \lambda^t f'(x^*) = 0$. Ce λ est le *multiplicateur de Lagrange* (ou de Kuhn et Tucker) associé au problème d'optimisation sous contraintes.

Cette méthode reste délicate d'emploi pour plusieurs raisons. D'abord, on ne va pas résoudre beaucoup de fois le problème pénalisé : c'est un travail potentiellement important. Donc quel ε choisir est non trivial. De plus, si u est trop "plat" au voisinage de C, son rôle dans u_ε va être masqué par le terme de pénalisation, nuisant à la précision de l'estimation de x^*. (Certes, l'hypothèse d'α-convexité limite ce risque en bornant inférieurement la courbure de u. Mais ceci souligne la dépendance de la méthode à cette hypothèse qui est d'habitude difficile à tester.)

Autres pénalisations On a pénalisé le critère avec la fonction de pénalisation $\|f^+\|^2$. Ceci pour avoir un critère u_ε dérivable. Le prix à payer est que, génériquement, si le minimum recherché est sur la frontière de C, on l'approchera par des points extérieurs. Les x_ε sont *non admissibles*.

On aurait pu prendre $f^+(x)$ tout simplement. Alors le critère n'est plus dérivable, mais il est possible qu'un ε fini permette déjà d'obtenir le minimum exact. Le caractère non différentiable du critère en x^* est néanmoins une grave difficulté. En fait, il vaut mieux se référer alors à la théorie de la dualité.

Pénalisation intérieure

Une autre approche possible est d'utiliser comme fonction de pénalisation une fonction "barrière", $\varphi(x)$, qui tend vers l'infini quand x tend vers la frontière de C par l'intérieur. On pourrait penser à $\varphi(x) = -\sum_i 1/\varphi_i(x)$ par exemple, mais nous verrons à la section suivante qu'un meilleur choix est $\varphi(x) = -\sum_i \ln(-f_i(x))$. Puis on va considérer $u_\varepsilon(x) = u(x) + \varepsilon\varphi(x)$, avec ε assez petit pour que ceci ne change guère le comportement du critère tant que les f_i sont tous loins d'être nuls. On parle alors d'algorithmes de "points intérieurs", parce qu'on aborde l'optimum recherché par des points intérieurs à C.

Soit donc d'une manière un peu plus générale, un critère perturbé $u_\varepsilon(x) = u(x) + \varepsilon\varphi(x)$ ou $\varphi(\cdot)$ est définie pour les x *intérieurs* au domaine des contraintes C (c'est à dire les x tels que $f_i(x) < 0$ pour $i = 1, \dots p$), convexe (continue) positive dans $\overset{\circ}{C}$, et tend vers l'infini quand $x \to \partial C$.

Alors, pour tout ε positif, u_ε atteint son minimum dans $\overset{\circ}{C}$, en un unique point x_ε dès lors que u est strictement convexe.

On a alors le résultat attendu :

Théorème 3.9 *Sous les hypothèses ci-dessus, et si u est une bonne fonction, $x_\varepsilon \to x^*$ quand $\varepsilon \to 0$.*

Demonstration Il faut faire attention qu'on ne peut pas manipuler $u_\varepsilon(x^*)$ parce que x^* peut être (est généralement) sur la frontière de C et donc φ peut n'y être pas définie. Soit donc δ un nombre positif (arbitrairement petit) et x_δ un point *itérieur* à C tel que $u(x_\delta) \leq u(x^*) + \delta$. La suite d'inégalités ci-dessous est facile à établir :

$$u(x_\varepsilon) \leq u_\varepsilon(x_\varepsilon) \leq u_\varepsilon(x_\delta) = u(x_\delta) + \varepsilon\varphi(x_\delta) \leq u(x^*) + \delta + \varepsilon\varphi(x_\delta)\,.$$

En prenant $\varepsilon \leq \delta/\varphi(x_\delta)$ on en déduit $u(x_\varepsilon) \leq u(x^*) + 2\delta$. Et comme δ était arbitraire, on en déduit que $u(x_\varepsilon)$ tend vers $u(x^*)$ quand ε tend vers zéro. Si u est α-convexe, on en déduit par utilisation de (1.16) que x_ε tend vers x^*.

Cette idée est à la base d'une méthode qui est aujourd'hui la meilleure connue si le problème est réellement convexe et si les dérivées secondes sont faciles à calculer. Nous la présentons ci-dessous.

3.3.5 Méthode du chemin central

Avant d'exposer cette utilisation de la pénalisation intérieure, montrons un résultat qui constitue la partie facile de la théorie de la dualité. Le problème considéré est toujours le même. Posons comme précédemment $\mathcal{L}(x,p) = u(x) + \sum_i p_i f_i(x)$.

Lemme 3.10 *Soit x^* la solution du problème d'optimisation sous contraintes inégalité. Soit $p \in \mathbb{R}^m$. On a*

$$\forall p \geq 0\,, \quad \min_x \mathcal{L}(x,p) \leq u(x^*)\,. \tag{3.9}$$

Démonstration On a la suite suivante d'inégalités faciles (la seconde vient de ce que pour $x \in C$, $\sum_i p_i f_i(x) \leq 0$) :

$$\min_x \mathcal{L}(x,p) \leq \min_{x\in C} \mathcal{L}(x,p) \leq \min_{x\in C} u(x) = u(x^*)\,.$$

Nous utilisons la méthode de pénalisation intérieure avec les fonctions barrières $-\ln(-f_i(x))$. Posons donc

$$u_\varepsilon = u - \varepsilon \sum_i \ln(-f_i(x))\,.$$

C'est une fonction fortement convexe. Notons x_ε son unique minimum sur $\mathbb{R}^n$. Nous affirmons le théorème facile, mais surprenant :

Théorème 3.11 *On a*

$$u(x_\varepsilon) - p\varepsilon \leq u(x^*) \leq u(x_\varepsilon)\,.$$

(Où p est ici le nombre de contraintes scalaires : $i = 1, \ldots, p$.)

Demonstration Le point x_ε est caractérisé par

$$\nabla u_\varepsilon(x_\varepsilon) = \nabla u(x_\varepsilon) - \varepsilon \sum_i \frac{1}{f_i(x_\varepsilon)} \nabla f_i(x_\varepsilon) = 0\,.$$

Posons alors $p_i = -\varepsilon / f_i(x_\varepsilon)$. Ce sont des nombres positifs. L'égalité ci-dessus s'écrit

$$\nabla u(x_\varepsilon) + \sum_i p_i \nabla f_i(x_\varepsilon) = \nabla_x \mathcal{L}(x,p) = 0\,.$$

Maintenant, ce $\mathcal{L}(x,p)$ est une fonction convexe de x. Donc la condition ci-dessus implique qu'elle atteint son minimum en x_ε. Utilisons alors le petit lemme précédent, en remarquant en outre que $p_i f_i(x_\varepsilon) = \varepsilon$. Le théorème en découle immédiatement.

Ainsi non seulement x_ε approche x^* quand $\varepsilon \to 0$, mais en outre on sait avec quelle précision $u(x^*)$ est atteint.

Voici quel est l'usage qu'on fait de ce résultat dans la méthode du chemin central. On commence typiquement avec $\varepsilon = 1$ (bien qu'une autre valeur soit bien sûr possible, et peut-être préférable dans certains cas.) On résoud le problème $\min u_\varepsilon$ sans contrainte par la méthode de Newton. La première application de cette méthode peut présenter ses difficultés habituelles et demander un peu de soin. Ensuite, on fait décroître ε, typiquement d'un ordre de grandeur à chaque fois, et on résoud à nouveau le problème sans contrainte en prenant la solution à l'étape précédente comme initialisation de la méthode de Newton. Cette fois on a une convergence très rapide de cette méthode. Et on continue à faire décroître ε jusqu'à ce qu'on ait la précision désirée. (On peut même améliorer le choix du x initial dans la méthode de Newton en extrapolant la courbe des x_ε antérieurs.)

On sait que le problème d'optimisation sans contrainte résolu à chaque itération par la méthode de Newton est de plus en plus mal conditionné au fur et à mesure que ε décroît. Mais pour une raison pas totalement expliquée, il semble que le choix d'initialisation de la méthode de Newton suggéré naturellement immunise l'algorithme contre les effets négatifs de ce mauvais conditionnement.

Cette méthode, due à Nesterov et Nemirovsky sur une idée de Karmarkar, puis améliorée par S. Boyd, a donné, entre les mains de ce dernier, des résultats spectaculaires sur de très nombreux problèmes. Elle est *la* méthode à appliquer …quand elle s'applique. (La convexité de u et des f_i est essentielle, et il faut que le calcul des dérivées secondes soit raisonnablement simple.)

3.4 Optimisation sous contraintes égalité

On considère à présent le cas où les contraintes sont de la forme

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid f_i(x) = 0\,, \quad i = 1, \ldots, p\}\,. \tag{3.10}$$

Il n'y a plus lieu de supposer une quelconque convexité pour les f_i ni u parce qu'on sort de toutes façons du cadre convexe. (C'est pour rappeler cela que nous n'avons pas utilisé la notation C —mais $\mathcal{C}$— pour l'ensemble des x admissibles.)

A quelques indications près, on laissera le soin au lecteur d'imaginer comment hybrider les algorithmes que nous allons discuter avec ceux du cas inégalité si une partie des contraintes est d'un type et une partie d'un autre.

3.4.1 Contraintes affines

On considère donc maintenant le cas où les f_i dans (3.10) sont affines, ou, de manière équivalente, il nous est donné une matrice F de type $p \times n$ et un vecteur f de dimension p, qui définissent les x admissibles comme devant vérifier

$$Fx = f \tag{3.11}$$

En outre, on supposera toujours que F est de rang p, donc surjective, ce qui impose en particulier que $p < n$. En effet, si ce n'est pas le cas, soit f est dans l'image de F, mais alors une partie des contraintes est redondante : on peut supprimer des lignes qui sont linéairement dépendantes des autres, soit f n'est pas dans l'image de F, et alors il n'y a pas de x admissible.

Algorithmes de gradient

La variété affine admissible est convexe. Donc l'algorithme de gradient projeté peut être conservé à l'identique. Il reste juste à faire remarquer que la projection est facile à calculer, au moins s'il n'y a pas trop de contraintes, et s'exprime par

$$P_{\mathcal{C}}(x) = (I - F^\dagger F)x + F^\dagger f \tag{3.12}$$

où

$$F^\dagger = F^t(FF^t)^{-1}$$

est une inverse à droite (l'inverse de Penrose) de la matrice surjective F.

En fait on peut faire mieux. En effet, la projection sur Ker F du gradient de u est alors le gradient de la restriction de u à $\mathcal{C}$. On peut donc appliquer un algorithme de gradient à pas optimal à cette restriction :

Algorithme Gradient projeté à pas optimal

1. Choisir une estimée initiale x^0, faire $k := 0$.
2. Calculer $\nabla u(x^k)$ et $h^k := (I - F^\dagger F)\nabla u(x^k)$
3. Si $\|h^k\| < \varepsilon$ stop. Si non,
4. Calculer $x^{k+1} := x^k - \theta^k h^k$ où le pas $\theta^k > 0$ est déterminé par

$$u(x^{k+1}) = \min_\theta u(x^k - \theta h^k)$$

5. incrémenter k de 1 et retourner en 2

Remarque 3.4 *Il est prudent d'ajouter une correction pour s'assurer que les x^k calculés restent bien dans la variété admissible, ce qui peut être perdu autrement en raison des erreurs d'arrondi. On peut intercaller à une fréquence à choisir un pas de projection sur $\mathcal{C}$ entre deux pas de gradient, soit le faire systématiquement à tous les pas, remplaçant la formule $x^{k+1} := x^k - \theta^k h^k$ par $x^{k+1} := (I - F^\dagger F)(x^k - \theta^k h^k) + F^\dagger f$.*

Les propriétés de convergence de cet algorithme se déduisent de son interprétation comme algorithme de gradient à pas optimal sur la restriction de u à $\mathcal{C}$.

Algorithme d'Uzawa

On a indiqué plus haut que le théorème de Kuhn et Tucker demeure pour des contraintes égalité affines, à ceci près que le signe des multiplicateurs n'est plus fixé.

En conséquence, l'algorithme d'Uzawa demeure, en supprimant l'opération de projection sur le cone positif. La preuve de convergence est inchangée.

Programmation quadratique

Un problème classique, dont on verra qu'il joue un rôle par la suite, est le problème d'optimiser une forme quadratique sous des contraintes affines. Il s'agit donc de minimiser

$$u(x) = \frac{1}{2} x^t Q x + q^t x$$

où Q est une matrice symétrique positive définie et q un vecteur de $\mathbb{R}^n$, sous les contraintes (3.11).

On laisse le lecteur démontrer (exercice) que la solution s'obtient conjointement avec le multiplicateur de Lagrange p en résolvant le système linéaire

$$\begin{array}{lcl} Qx + F^t p & = & -q\,, \\ Fx & = & f \end{array} \tag{3.13}$$

Théorème 3.12 *Le systèmes d'équations (3.13) a une solution unique quelque soit F si et seulement si F est surjective et la restriction de Q au noyau de F est définie (positive ou négative).*

Preuve Il suffit de vérifier si le système *homogène*, c'est à dire obtenu en remplaçant q et f par 0, a zéro pour seule solution. Soit donc (x, p) satisfaisant le système homogène. Multiplions la première équation à gauche par x^t et tenons compte de la deuxième pour constater qu'on doit avoir $x^t Q x = 0$. Mais ce x appartient nécessairement au noyau de F. Donc $x^t Q x = 0$ implique $x = 0$ si et seulement si la restriction de Q à ce noyau est définie. Mais dans ce cas, on doit aussi avoir $F^t p = 0$, qui implique $p = 0$ si et seulement si F est surjective.

Si Q est inversible, le système (3.13) admet la solution

$$x^* = Q^{-1} F^t (F Q^{-1} F^t)^{-1} (F Q^{-1} q + f) + Q^{-1} q\,.$$

Cependant, demander l'inversibilité de Q est trop, puisque seule doit être positive définie sa restriction au noyau de F. On peut donner une formule explicite qui n'utilise que cette hypothèse, en fonction de la décomposition en valeurs singulières de F :

$$F = U \Sigma V = U [\Sigma_1 \quad 0] \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

(où U et V sont des matrices orthogonales de type $p \times p$ et $n \times n$ respectivement et Σ_1 est la matrice diagonale des valeurs singulières, V_2^t engendre Ker F) comme

$$x^* = (I - V_2^t(V_2QV_2^t)^{-1}V_2Q)V_1^t\Sigma_1^{-1}U^t f - V_2^t(V_2QV_2^t)^{-1}V_2q\,.$$

En pratique, on a construit des algorithmes d'élimination très spécialisés, plus rapides que le calcul d'une décomposition en valeurs singulières suivi de l'inversion de $V_2QV_2^t$.

3.4.2 Contraintes nonlinéaires

Nous sommes maintenant dans les notations de (3.10).

Algorithme à la Uzawa

Le théorème de Kuhn et Tucker n'est plus valide si les contraintes ne sont pas affines. On peut tenter d'appliquer encore l'algorithme d'Uzawa. On n'est assuré ni de sa convergence, ni du fait que s'il converge ce soit vers l'optimum. Voici une brève analyse de cette question.

Remarquons que la fonction étendue

$$\varphi(x) = \sup_\lambda(\lambda, f(x))$$

vaut 0 si x est admissible, et $+\infty$ si non. Ainsi le problème posé, de minimiser $u(x)$ sous les contraintes $f(x) = 0$, est équivalent au problème de minimiser $u(x) + \varphi(x)$, soit encore de chercher

$$\min_{x\in\mathbb{R}^n} \sup_\lambda(u(x) + (\lambda, f(x)))\,.$$

On a encore un minsup, et ceci justifie qu'on fasse un algorithme de type gradient ascendant en λ. Mais ce que calcule l'algorithme d'Uzawa, c'est le

$$\max_\lambda \min_{x\in\mathbb{R}^n} (u(x) + (\lambda, f(x)))\,.$$

En général, ces deux quantités sont différentes, la deuxième inférieure à la première, ce qu'on appelle le "saut de dualité".

Cette méthode ne doit donc être tentée qu'avec circonspection, et si elle converge il convient de vérifier si f s'annulle au point trouvé. (Ce qui n'est garanti dans le cas convexe que par le fait que $\min_x \sup_p = \max_p \min_x$.)

Programmation quadratique séquentielle

Si les conditions requises sont satisfaites, c'est à dire si les $\nabla f_i(x^*)$ sont linéairement indépendants, le point recherché est caractérisé par le théorème de Lagrange. C'est à dire qu'il est, avec le multiplicateur (vectoriel) de Lagrange λ, solution du système

$$\begin{array}{lcl} \nabla u(x) + \nabla f(x)\lambda & = & 0\,, \\ f(x) & = & 0\,. \end{array} \tag{3.14}$$

Ici, $\nabla f(x)$ désigne la matrice jacobienne de f transposée $(f'(x))^t$, de même que $\nabla u(x)$ désigne le transposé de $u'(x)$.

Une idée naturelle, et fructueuse, consiste à essyer de résoudre ce système d'équations non linéaires par la méthode de Newton.

Nous introduisons quelques notations à cet effet. On notera $Q := D_x^2(u(x)+(\lambda, f(x)))$ la matrice symétrique des dérivées secondes en x du lagrangien, et $F := f'(x)$ la matrice jacobienne de f. En outre, pour alléger encore les notations, on notera $f^k := f(x^k)$ et de même pour toutes les fonctions de x et λ.

Avec ces notations, l'algorithme de Newton s'écrit

$$\begin{array}{lcl} Q^k x^{k+1} + (F^k)^t \lambda^{k+1} & = & Q^k x^k - \nabla u^k \,, \\ F^k x^{k+1} & = & F^k x^k - f^k \,. \end{array}$$

On remarque que ce système, où les membres de droite sont connus à l'étape k, et les inconnues sont x^{k+1} et λ^{k+1}, a exactement la même forme que le système (3.13). Il peut donc être résolu à l'aide un algorithme de programmation quadratique —d'où le nom de cette méthode.

Ajoutons que la théorie de la seconde variation montre que les conditions énoncées au théorème 3.12 sont excatement ici la condition de qualification des contraintes d'une part, et la condition suffisante du deuxième ordre pour un minimu local sous contraintes d'autre part.

On peut donc énoncer le théorème :

Théorème 3.13 *Si l'optimum x^* recherché existe, et si la condition de qualification des contraintes y est satisfaite ainsi que la condition suffisante locale du deuxième ordre, il existe un voisinage de x^* dans lequel l'algorithme de programmation quadratique séquentielle converge.*

Bien sûr, avec ses qualités —très grande vitesse de convergence—, cet algorithme partage les défauts de la méthode de Newton : faible robustesse, et nécessité de ce fait de partir avec une assez bonne estimée de x^*.

Chapitre 4

Programmation linéaire et programmation dynamique

Les deux sujets qu'effleure ce chapitre, à titre d'introduction, ont en commun de se situer à la frontière de l'optimisation continue et de l'optimisation combinatoire.

La programmation linéaire parle de variables continues sous des contraintes continues, mais le premier pas dans l'étude de ce problème est de montrer qu'un nombre fini de points sont candidats à être optimaux, et l'algorithme du simplexe peut être vu comme une façon habile de parcourir cet ensemble fini.

De son côté, la programmation dynamique sera d'abord présentée comme un problème de recherche du plus court chemin dans un graphe, donc fondamentalement combinatoire. Mais on verra ensuite qu'un procédé classique ramène un problème en variable d'espace continue (et variable de temps discrète) à un tel graphe.

4.1 Programmation linéaire

4.1.1 Position du problème

Bien des modèles en ingéniérie et en économie mènent à considérer des problèmes où critère et contraintes sont linéaires (donc pas α-convexe) et les variables positives. Ces problèmes ont reçu le (vieux) nom de "programmes linéaires". Ils ont été étudiés dès les origines de ce qu'on devait appeler la "recherche opérationnelle", notamment par G.B. Dantzig dès le début des années 1950 (le plus ancien article cité remonte à 1951), et le développement des méthodes numériques afférentes est contemporain de l'apparition des ordinateurs.

Comme modèles simplifiés très naturels de situations concrètes (coûts et ressources consommées proportionnels aux quantités), ces problèmes ont justifié un effort algorithmique considérable, et ce jusque dans les années plus récentes, comme le bruit fait par les Bell labs autour de la "méthode de Karmarkar" l'a montré.

Nous nous contenterons ici de donner un aperçu des résultats de base et des idées sous-jacentes à l'algorithme du simplexe, comme introduction à l'utilisation des programmes existants. En effet, écrire un nouveau programme de programmation linéaire, que ce soit par l'algorithme du simplexe ou une autre méthode, est un exercice *strictement réservé aux professionnels* de la chose, tant les "packages" qui existent sont (nombreux et) perfectionnés.

On utilisera les inégalités entre vecteurs

$$x \geq y \quad \Leftrightarrow \quad x_i \geq y_i \,,\ i = 1, \ldots,$$

$$x > y \quad \Leftrightarrow \quad x \geq y \text{ et } x \neq y \,,$$

$$x >> y \quad \Leftrightarrow \quad x_i > y_i \,,\ i = 1, \ldots \,.$$

Le problème peut en général être formulé de la façon suivante.

Le critère à minimiser est déterminé par un vecteur c de $\mathbb{R}^n$, et est donné par

$$u(x) = (c, x) = \sum_{i=1}^{n} c_i x_i$$

Les contraintes sont d'une part

$$x \geq 0 \,,$$

et d'autre part deux jeux de contraintes définies par deux matrices A et B, de dimensions respectives $p \times n$ et $q \times n$, et deux vecteurs a et b de dimension p et q définissant les contraintes égalité et inégalité par

$$Ax = a \,, \qquad Bx \leq b \,.$$

Dans les discussions qui suivent, il faut avoir à l'esprit que n, p et q peuvent être de l'ordre de plusieurs milliers, voire dizaines de milliers.

La première remarque est qu'au moins au plan théorique, on peut remplacer les contraintes inégalité par des contraintes égalité et inversement par les artifices suivants. En introduisant q variables supplémentaires $y \in \mathbb{R}^q$, on peut remplacer $Bx \leq b$ par

$$Bx + y = b \,, \quad y \geq 0 \,.$$

Au contraire, si on veut privilégier les contraintes inégalité, on peut remplacer $Ax = a$ par

$$Ax \leq a \text{ et } Ax \geq a \,.$$

Bien sur, la première de ces opérations *augmente de q le nombre de variables*, tandis que la deuxième *augmente de p le nombre de contraintes*. Deux opérations qui ne sont pas souhaitables au vu des dimensions que nous évoquions. Ce sont par contre des artifices utiles pour étudier les propriétés théoriques du problème.

Nous utiliserons dans l'étude théorique la *forme standard égalité* caractérisée par m contraintes égalité *signées* :

$$Ax = b \,, \qquad b \geq 0 \,,$$

ce qui est toujours possible, quitte à multiplier certaines contraintes par -1, et toujours les contraintes de positivité

$$x \geq 0 \,.$$

4.1.2 Étude du polyèdre

Pour comprendre le problème de programmation linéaire, il faut étudier l'ensemble des points défini par les contraintes linéaires et de positivité, appelé *polyèdre*, que nous prenons sous la forme standard égalité $Ax = b$. Nous noterons P cet ensemble.

Nous appelons encore n la dimension de x, sachant qu'elle a pu être augmentée pour donner cette forme aux contraintes. Nous appelons m le nombre de contraintes, et supposons que $m < n$. Nous supposons même plus que cela, à savoir que

$$\text{rang} A = m\,. \tag{4.1}$$

En effet, si-non A a des lignes linéairement dépendantes d'autres, et soit la même dépendance linéaire se retrouve dans les coordonnées de b, et cette ligne est surnuméraire, et peut être omise sans rien changer au problème, soit les coordonnées de b n'exhibent pas la même dépendance, et le polyèdre est vide.

Nous introduisons à cette fin la terminologie suivante :

Définition 4.1 (Direction de $\mathbb{R}^n$) *Nous appelons* direction *l'ensemble des vecteurs portés par une même demi-droite, c'est à dire multiples positifs d'un d'entre-eux.*

Ainsi, soit $h \in \mathbb{R}^n$, il définit une direction $\{\theta h \mid \theta \in \mathbb{R}_+\}$, et tout vecteur de cette direction définit la même direction.

Une direction sera réputée "admissible" si elle est composée de vecteurs à coordonnées positives (ce qui est le cas dès qu'un de ses vecteurs l'est) et si elle appartient au noyau de A. Ainsi, si $x \in \mathsf{P}$, $x + w \in \mathsf{P}$ pour tout w dans cette direction. Ce sont des directions dans lesquelles P est non borné. (Très précisément, ces directions définissent des "points à l'infini" du polyèdre au sens de la géométrie projective.)

Le résultat essentiel que nous visons est le suivant :

Théorème 4.1 *Les points d'un polyèdre peuvent tous être obtenus comme combinaison convexe d'un nombre fini de ses points (appelés* sommets*) plus une somme d'éléments d'un nombre fini de directions (appelées* directions admissibles extrêmales *ou* sommets à l'infini*). Réciproquement, toute combinaison de cette forme appartient au polyèdre.*

Ainsi tout polyèdre est caractérisé par un nombre fini de sommets, à distance finie ou à l'infini, et est constitué par l'ensemble de leurs combinaisons convexes. En particulier, s'il est borné, un polyèdre se réduit au *polytôpe* de toutes les combinaisons convexes de ses sommets (en nombre fini).

Pour démontrer ce résultat, nous introduisons la définition suivante :

Définition 4.2 (Points extrémaux) *Étant donné un ensemble convexe* C*, on appelle* points extrémaux *de* C *les points de* C *qui ne peuvent être représentés comme une combinaison convexe propre d'autres points de* C.

Ainsi, si $\hat{x}$ est un point extrémal de C, et si x_1 et x_2 sont deux points de C tels que $\hat{x} = \lambda x_1 + (1 - \lambda)x_2$, alors nécessairement $\lambda = 0$ ou $\lambda = 1$, et $\hat{x}$ coincide avec x_1 ou x_2.

La même définition s'appliquera aux directions admissibles, sachant que deux vecteurs colinéaires et de même sens (proportionnels dans un rapport positif) représentent la même direction.

Lemme 4.2 *Un point admissible (i.e. tel que $x \geq 0$ et $Ax = b$) [resp une direction admissible h, i.e. telle que $h \geq 0$ et $Ah = 0$] est extrémal[e] si et seulement si les colonnes de A correspondant aux coordonnées* non nulles *de x [resp h] sont linéairement indépendantes [resp. ont un défaut de rang égal à 1].*

Les coordonnées non nulles de x sont dites "de base". On notera que la propriété ci-dessus implique notamment qu'un point extrémal a au plus m coordonnées de base.

Démonstration du lemme Pour simplifier l'écriture, nous supposons que ce sont les p premières colonnes de A qui sont concernées, et donc les $n - p$ dernières coordonnées de x qui sont nulles. Nous partitionnons A et x en

$$A = [\bar{A}\ \tilde{A}]\,, \qquad \begin{pmatrix} \bar{x} \\ \tilde{x} \end{pmatrix}$$

Soit un point de P qui satisfait la condition énoncée, c'est à dire que $x = (\bar{x} \quad 0)$, et donc $\bar{A}\bar{x} = b$. Si ce point est combinaison convexe propre de deux autres points de P alors, il est intérieur au segment joignant ces deux points, ce qui veut dire qu'il existe un vecteur $w \neq 0$ tel que $x + w$ et $x - w$ appartiennent à C. En particulier, ceci implique que les composantes de w hors base soient nulles (si non, soit $x + w$ soit $x - w$ aurait des composantes négatives), et donc aussi que $\bar{A}\bar{w} = 0$, où $\bar{w}$ désigne bien sûr les m premières coordonnées de w. Mais par hypothèse, $\bar{A}$ est injective, donc $w = 0$, ce qui contredit l'hypothèse.

Réciproquement, soit x un point de P, $\bar{x}$ l'ensemble de ses coordonnées non nulles, que nous regroupons au début de la numérotation, et $\bar{A}$ la matrice des colonnes corrsepondantes. Si les colonnes de $\bar{A}$ ne sont pas linéairement indépendantes (A n'est pas injective), il existe un vecteur $\bar{w}$ non nul tel que $\bar{A}\bar{w} = 0$. Comme les coordonnées de $\bar{x}$ sont toutes strictement positives, il existe ε assez petit pour que, en choisissant $\|\bar{w}\| \leq \varepsilon$ ce qui est toujours loisible, les coordonnées de $\bar{x} + \bar{w}$ et de $\bar{x} - \bar{w}$ soient encore toutes positives. Alors, en complètant $w = (\bar{w} \quad 0)$, on voit que $Aw = 0$, et que $x + w$ et $x - w$ appartiennent tous les deux à C. Donc $x = 1/2(x + w) + 1/2(x - w)$ n'est pas un point extrémal de P.

On laisse le lecteur répéter la même preuve pour les directions admissibles extrémales, en se souvenant que la différence entre le nombre des vecteurs considéré et le rang du système qu'ils forment, ou *défaut de rang*, est la dimension du noyau de la matrice dont ils sont les colonnes.

Cette caractérisation montre qu'il ne saurait y avoir qu'un nombre fini de points extrémaux à P. Il suffit de tester tous les ensembles de m colonnes de A, et pour ceux qui sont linéairement indépendants, de verifier si $A^{-1}b$ est positif. On fera de même avec les ensembles de $m + 1$ colonnes de rang m pour chercher les directions admissibles optimales.

Démonstration du théorème Soit x un point de C, et supposons que ce ne soit pas un point extrémal. Comme précédement, il existe un vecteur w du noyau de A ayant les mêmes coordonnées nulles que x, et tel que $x - w$ et $x + w$ appartiennent à C. Regardons $x + tw$, $t > 0$. Si w n'est pas une direction admissible de C (il a des coordonnées négatives), pour un certain t^+ une des coordonnées de ce vecteur s'annule, les autres étant encore positives. On peut faire de même avec $x - tw$. (Si w est une direction admissible de C, $-w$ ne l'est pas, et réciproquement.) Donc, soit t^+ et t^- sont tous les deux définis, et x peut être représenté comme une combinaison convexe de deux vecteurs qui ont une coordonnée de plus nulle chacun : $x = t^-/(t^+ + t^-)x^+ + t^+/(t^+ + t^-)x^-$, soit on a une représentation comme une somme $x = x^- + t^-w$ où w est une direction admissible, et x^- a une coordonnée nulle de plus que x. En répétant ce processus pour les éléments de cette représentation récursivement, on obtient le théorème. (On ne fait qu'un nombre fini de fois cette opération, puisque le nombre de coordonnées

non nulles diminue chaque fois, et la construction s'arrête quand A n'a plus de noyau —ou un noyau de dimension 1 pour les directions admissibles—.)

On déduit de ce théorème le corollaire fondamental suivant :

Corollaire 4.3 *Si le polyèdre des contraintes n'est pas vide, soit le critère n'a pas d'infimum fini, soit un des points extrémaux (ou sommets) du polyèdre est solution.*

Démonstration Soit il existe une direction admissible w telle que $(c, w) < 0$. Alors, comme on peut ajouter tw, t positif arbitraire, à tout point du polyèdre sans en sortir, clairement, le critère (c, x) peut être rendu arbitrairement grand négatif. Soit, pour toute direction admissible w, $(c, w) \geq 0$. Alors, si un point du polyèdre a w dans sa décomposition comme au théorème ci-dessus, on peut retirer cette composante. On reste dans le polyèdre, et on améliore le critère. Nous ne considérons donc que des points combinaison convexe des sommets x_1 à x_N :

$$x = \sum_{i=1}^{N} \lambda_i x_i, \qquad \lambda_i \geq 0, \qquad \sum_{i-1}^{N} \lambda_i = 1\,.$$

Alors,

$$(c, x) = \sum_{i=1}^{N} \lambda_i (c, x_i)\,.$$

Si le critère (c, x) est minimum sur C, tous les (c, x_i) sont supérieurs ou égaux à (c, x). Donc seuls peuvent être non nuls les λ_i pour lesquels on a l'égalité, et ce sont là des sommets solution. Ou bien — et c'est le cas générique— un seul sommet est solution, et il ne peut être représenté comme ci-dessus de manière non dégénérée.

4.1.3 L'algorithme du simplexe

Nous ne donnons ici qu'un bref aperçu du principe de l'algorithme le plus célèbre pour résoudre numériquement le programme linéaire, l'algorithme du simplexe. Il en existe d'autres aujourd'hui, plus rapides ... dans la plupart des configurations.

Remarquons d'abord qu'en principe, puisque le nombre de sommets est fini et qu'on sait les déterminer tous, il suffit de comparer la valeur du critère à tous ces sommets et prendre le meilleur. Ce qui s'oppose à cette approche naïve est le nombre potentiellement très grand de sommets du polyèdre. On considère couramment des problèmes où n et m sont tous les deux en milliers. Or il y a $n!/m!(n-m)!$ combinaisons de m colonnes à tester. C'est à dire, si $n = 2000$ et $m = 1000$, un nombre de l'ordre de 10^{600} combinaisons.

Il faut faire autrechose. L'algorithme du simplexe va explorer des sommets d'une façon moins naïve, et d'habitude plus efficace. On sait malheureusement exhiber des problèmes pour lesquels cet algorithme explore *tous* les sommets. Mais on sait que ce n'est *génériquement* pas le cas, et que le simplexe est génériquement polynomial en $m + n$.

Le principe de l'algorithme est donc le suivant. Étant donné le choix d'une "base", c'est à dire de m colonnes indépendantes de A formant $\bar{A}$, de sorte que $A = [\bar{A}\ \tilde{A}]$, et un découpage correspondant des coordonnées de tout x en coordonnées en base $\bar{x}$ et hors base $\tilde{x}$, on a nécessairement

$$\bar{A}\bar{x} + \tilde{A}\tilde{x} = b\,,$$

soit encore

$$\bar{x} = -\bar{A}^{-1}\tilde{A}\tilde{x} + \bar{A}^{-1}b\,. \tag{4.2}$$

Donc, en paramétrisant x via $\tilde{x}$:

$$(c, x) = (\tilde{c}^t - \bar{c}^t \bar{A}^{-1} \tilde{A})\tilde{x} + \bar{c}^t \bar{A}^{-1} b \,,$$

que nous réecrivons avec une notation évidente

$$(c, x) = (\tilde{w}, \tilde{x}) + \bar{c}^t \bar{A}^{-1} b \,.$$

Si l'itérée x^k de l'algorithme est un sommet correspondant à cette base, soit $\tilde{x}^k = 0$, on va chercher à améliorer le critère en repérant la coordonnée de $\tilde{w}$ la plus négative, disons $\tilde{w}_M$ et en donnant une valeur positive à la coordonnée $\tilde{x}_M$ correspondante de x. Soit donc e_M le vecteur de base numéro M de $\mathbb{R}^n$, et essayons des x de la forme

$$x = x^k + t e_M \,.$$

On est sûr de faire décroitre le critère ce faisant. Le t maximum permis est atteint la première fois qu'une autre coordonnée de $\bar{x}$, tel que calculé par (4.2) passe par zéro. On s'arrête à cette valeur de t, on fait ainsi rentrer la coordonnée M dans la base et sortir celle qui s'est annulée. Et on itère.

L'algorithme s'arrête quand toutes les coordonnées de $\tilde{w}$ sont positives : on ne peut plus améliorer le critère.

Il reste à dire comment *initialiser* l'algorithme. En effet, nous avons indiqué comment passer d'un sommet du polyèdre à un autre, mais comment trouver un premier sommet ? Nous allons indiquer comment utiliser le même algorithme pour trouver un sommet initial, et en même temps déterminer s'il existe un tel sommet, c'est à dire si l'ensemble des états admissibles est non vide. Il peut en effet se faire que les contraintes $Ax = b$, $x \geq 0$ soient incompatibles, mais ceci même est difficile à découvrir.

La méthode va consister à examiner un autre problème linéaire qui présente la particularité d'avoir un sommet évident, et de chercher, s'il existe, un sommet du problème d'origine.

Supposons qu'on s'est ramené à $b \geq 0$, ce qu'on peut toujours faire en changeant le signe des lignes de A si nécessaire. Considérons le problème de programmation linéaire portant sur les variables (positives) (x, w), $x \in \mathbb{R}^n$, $w \in \mathbb{R}^m$, dont les contraintes sont

$$Ax + w = b$$

et le critère

$$u(x, w) = \sum_{i=1}^{m} w_i \,.$$

Comme promis, les contraintes admettent une solution évidente, qui est un sommet : $x = 0$, $w = b$. Et comme ce critère est toujours positif ou nul, son optimum sera zéro si et seulement s'il existe une solution des contraintes avec $w = 0$, soit un point admissible du problème d'origine. En outre, évoluant de point extrémal en point extrémal, le simplexe nous donnera un point extrémal, qui pourra à son tour servir d'initialisation pour le problème d'origine.

Il y a beaucoup à dire à partir de là (on a écrit des livres entiers). Que faire dans le simplexe si la nouvelle base n'est pas indépendante ? si deux coordonnées s'annulent en même temps ? etc. On laisse le lecteur imaginer des parades simples à ces questions.

Plus intéressant : comment simplifier le calcul de $\bar{A}^{-1}$ pour la nouvelle base en utilisant le fait qu'on connaissait cette inverse pour une matrice ne différant que par une colonne ?

Pour toutes ces questions, nous renvoyons le lecteur aux livres spécialisés.

4.1.4 Rudiments de dualité

On ne peut pas parler de programmation linéaire sans évoquer la dualité, qui constitue, comme on va le voir, un outil très utile.

Dans ce numéro, et pour l'élégance des formules obtenues, nous allons supposer qu'on cherche à maximiser un critère linéaire $u = (c, x)$. Cela revient évidemment à changer c en $-c$, mais comme nous n'avons jamais fait d'hypothèse sur le signe des éléments de c, cela est sans conséquence.

Partons donc d'un problème standard, que nous écrivons

$$\max_x c^t x\,, \qquad x \geq 0\,, \qquad Ax = b\,,$$

en notant de façon explicite $c^t x$ le produit scalaire (c, x). Supposons que nous connaissions un vecteur y de $\mathbb{R}^p$ (p est le nombre de contraintes) tel que

$$y^t A \geq c^t\,.$$

Alors, pour tout $x \geq 0$, on a $c^t x \leq y^t A x$. Mais par ailleurs, pour tout x admissible, $Ax = b$, de sorte que l'inégalité ci-dessus donne

$$c^t x \leq y^t b\,. \tag{4.3}$$

Avant d'aller plus loin, supposons que la contrainte $Ax = b$ provient en fait d'une contrainte inégalité, et qu'on a augmenté l'état de "variables d'écart" pour en faire des contraintes égalité, c'est à dire, avec un abus de notations évident, que le vecteur x ci-dessus est en fait de la forme

$$x = \begin{pmatrix} x \\ \xi \end{pmatrix}\,,$$

que A est donc de la forme

$$A = [\ A \quad I\ ]\,,$$

de sorte que la contrainte est

$$Ax + \xi = b, \qquad \xi \geq 0, \qquad x \geq 0\,,$$

donc équivalente à

$$Ax \geq b\,, \qquad x \geq 0\,. \tag{4.4}$$

La "contrainte" sur y se lit alors

$$y^t[\ A \quad I\ ] \geq [\ c^t \quad 0\ ]$$

soit

$$y^t A \geq c^t\,, \qquad y \geq 0 \tag{4.5}$$

Ainsi, pour tout x admissible au sens de (4.4) et tout y admissible au sens du *problème dual* dont l'admissibilité est définie par (4.5), on a (4.3). Le problème

$$\min_y (b, y)$$

soumis aux contraintes (4.5) est appelé *problème dual* de celui d'origine (rappelons que nous en avons ici fait un problème de maximisation sous contraintes inégalité).

On remarque la parfaite symétrie entre problème primal et dual, de sorte que toute affirmation concernant leur intéraction peut être énoncée dans un sens ou dans l'autre.

On voit que si on trouve x et y admissibles pour les problèmes primal et dual tels que $(c, x) = (b, y)$, alors ils sont nécessairement solution de ces problèmes. Cette remarque est la base de méthodes efficaces pour résoudre le problème de programmation linéaire, visant à minimiser la différence, toujours positive ou nulle, $(b, y) - (c, x)$ parmi les x et y admissibles.

On voit aussi aisément que si le problème dual a un ensemble d'états admissibles non vide, le problème primal a son critère borné supérieurement (et mutatis mutandis pour le problème dual si le problème primal admet des états admissibles).

Ces constatations élémentaires ont des réciproques, que nous ne démontrerons pas :

Théorème 4.4 *Chacun des deux problèmes, primal et dual, a un ensemble d'états admissibles non vide et un suprémum fini (donc une solution) si et seulement si il en va de même de l'autre. Si un des deux problèmes a un extremum infini, l'autre n'a pas d'état admissible.*

Donc les deux problèmes ont une solution ou n'en ont pas simultanément, l'absence de solution pouvant provenir de l'absence d'états admissibles, ou du fait que le critère n'est pas borné.

Théorème 4.5 *Si les problèmes de programmation linéaire primal et dual ont une solution, il existe x^* et y^* tels que $(c, x^*) = (b, y^*)$. Réciproquement, toute paire admissible (x^*, y^*) satisfaisant cette égalité est optimale.*

Enfin, faisons remarquer que la connaissance de y^*, par exemple, permet de trouver x^*. En effet, remarquons qu'on a donc toujours, pour tout x et y admissibles

$$c^t x \leq y^t A x \leq y^t b\,,$$

de sorte que si $c^t x = y^t b$, non seulement x et y coincident avec les optimums, mais aussi, toutes les inégalités ci-dessus sont des égalités. Or, l'égalité

$$(y^*, Ax^* - b) = 0$$

alors que $y^* \geq 0$ et $Ax - b \leq 0$ montre que pour toute coordonnée de y^* non nulle, la contrainte correspondante en x est "saturée" en x^* (satisfaite avec l'égalité). De même, l'égalité $(c^t - (y^*)^t A)x^* = 0$, que nous pouvons réécrire

$$(A^t y^* - c, x^*) = 0$$

alors que $A^t y^* - c \geq 0$ et $x^* \geq 0$ implique que pour toute contrainte duale non saturée, la coordonnée corrsepondante de x^* est nulle. Comme nous l'avons vu plus tôt, connaître les contraintes saturées (les ξ_i nuls) et les x_i nuls suffit à déterminer x^* par l'équation linéaire qu'on en déduit.

L'utilité de cette théorie est double. D'une part, comme nous l'avons indiqué, elle est le fondement de méthodes numériques efficaces, ou d'améliorations de l'algorithme du simplexe. D'autre part, elle permet toujours de choisir si l'on préfère résoudre le problème primal ou dual. L'un a plus de contraintes (inégalité) que de variables, et l'autre plus de variables que de contraintes. Suivant les packages de résolution disponibles, l'un peut être préférable à l'autre.

4.2 Programmation dynamique

Nous allons partir d'une vision combinatoire de la programmation dynamique, pour aboutir à une utilisation dans des problèmes authentiquements "dynamiques" en variables continues, donc voisins des préoccupations du reste de ce cours.

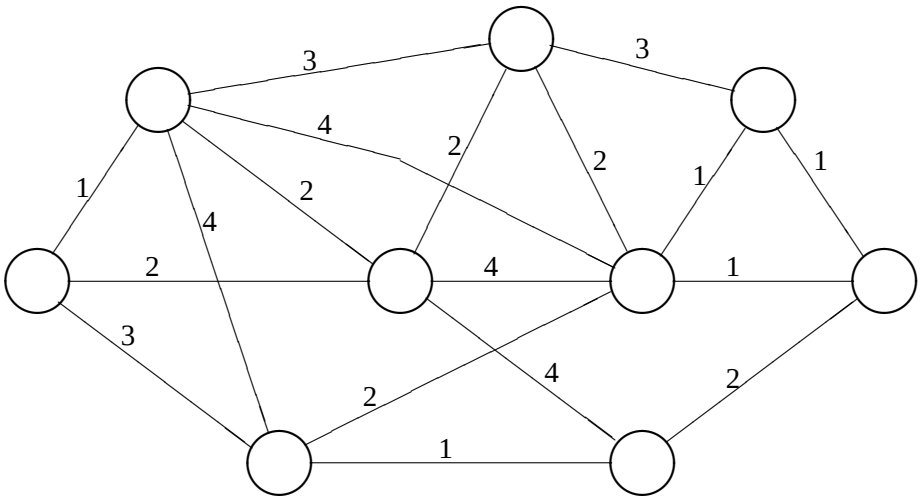

FIG. 4.1 – Graphe et “poids” des arcs

4.2.1 Plus court chemin dans un graphe orienté

Le problème le plus simple

Nous considérons un graphe orienté, ici de gauche à droite, comme celui de la figure 1, dont chaque arc est muni d’une “longueur” ou “poids”.

Le problème posé est de trouver le chemin de “longueur” ou “poids” minimum du nœud initial, le plus à gauche, au nœud terminal, le plus à droite. C’est manifestement un problème “fini” : le graphe ne comporte qu’une vingtaine de chemins. On pourrait donc tous les lister et choisir le plus court. Mais on voit bien que le nombre de chemins croit combinatoirement avec la taille du graphe, et une procédure aussi rustique ne s’étendra pas à des situations beaucoup plus complexes. (Le seul travail de répertorier tous les chemins devient exorbitant.)

Nous allons indiquer une procédure qui ne croit que linéairement avec le nombre de nœuds, ou plus précisément comme le produit du nombre de nœuds par le nombre moyen d’arcs par nœud.

Le principe de la méthode est de marquer chaque nœud avec la longueur du chemin minimal de ce nœud jusqu’à la fin. Cette procédure sera rapide en raison de la remarque banale mais essentielle qui suit :

Proposition 4.6 (Principe de Bellman) *Le chemin optimal a la propriété (dite “principe d’optimalité” de Bellman)*[1] *qu’entre tout nœud $\mathcal{N}$ par où il passe et la fin du chemin, il est optimal pour le problème d’aller de ce nœud $\mathcal{N}$ jusqu’à la fin. (Ce que nous appellerons le* sous-problème initialisé en $\mathcal{N}$*.)*

Démonstration Comme la longueur totale du chemin est la somme de la longueur parcourue du nœud initial au nœud $\mathcal{N}$ plus la longueur de $\mathcal{N}$ jusqu’à la fin, si on pouvait trouver un chemin plus court pour cette dernière longueur, —le sous problème initialisé en $\mathcal{N}$— on pourrait, en le concaténant avec

[1] Il s’agit de Richard Bellman, dont on peut contester qu’il ait inventé la programmation dynamique, pas qu’il en ait compris le premier toute la puissance et toute la généralité.

le chemin optimal entre le début et $\mathcal{N}$, trouver un chemin global plus court que le chemin optimal, ce qui est une contradiction.

Cette démonstration semble être une tautologie tant la propriété est évidente. Pourtant, nous allons nous servir de ce “principe de Bellman” en le reformulant un peu.

Pour tout nœud, *si* le chemin optimal passe par ce nœud, de ce nœud jusqu'à la fin, il utilise le chemin optimal pour ce sous problème. En particulier, depuis un nœud $\mathcal{N}$, si la longueur du chemin optimal jusquà la fin —c'est à dire la *valeur* optimale des sous problèmes si non leur solution complète— est connue pour tous les nœuds immédiatement aval (c'est à dire séparés par un seul arc), alors résoudre le sous problème initialisé en $\mathcal{N}$ est immédiat. En effet, *si* le chemin optimal (depuis $\mathcal{N}$) passe par un certain $\mathcal{N}'$ aval, la longueur en est la somme de la longueur de l'arc séparant $\mathcal{N}$ de $\mathcal{N}'$ ajoutée à la valeur optimale du sous-problème initialisé en $\mathcal{N}'$. Ainsi, depuis $\mathcal{N}$ il suffit de comparer ces valeurs, et de retenir la meilleure (la plus petite). On aura ainsi à peu de frais la valeur du sous-problème initialisé en $\mathcal{N}$.

On obtient ainsi l'algorithme suivant :

Algorithme Programmation dynamique simple

1. Marquer le nœud terminal avec la valeur 0.
2. En tout nœud dont tous les nœuds immédiatement aval sont déjà marqués, faire :
   – pour chaque nœud immédiatement aval, calculer la somme de la longueur de l'arc vers ce nœud et de la valeur de ce nœud.
   – Prendre la plus petite de ces sommes pour valeur du nœud courant, et le marquer.
   – Marquer le, ou les, arc(s) donnant la valeur retenue.
3. Retourner en (2) jusquà ce que tous les nœuds soient marqués
4. depuis le nœud initial, (comme depuis tout nœud du graphe) tout chemin n'empruntant que des arcs marqués est optimal, et a une longueur égale à la valeur marquée à ce nœud.

À titre d'exemple, nous avons dans la figure 2 marqué à chaque nœud sa valeur, et renforcé les arcs optimaux. On voit que le problème posé n'avait pas une solution unique, mais cette procédure n'en est nullement affectée.

Extensions du problème du plus court chemin

On peut étendre de nombreuses façons cet algorithme. La plus élémentaire est la suivante. On peut considérer *plusieurs* nœuds terminaux et plusieurs nœuds initiaux possibles. De plus, on peut supposer qu'un “poids” est attaché à chacun des nœuds en plus des arcs.

Dans ce dernier cas, on pourrait aussi bien attacher ce poids du nœud à tout arc qui le rejoint, se ramenant de manière triviale au problème où seuls les arcs ont un poids. On préfère, pour des raisons qui apparaîtront plus loin, considérer d'une part le poids attaché à chaque nœud terminal, et considérer que c'est la *valeur* de ce nœud pour l'algorithme de programmation dynamique, et associer les poids des autres nœuds aux arcs qui les quittent.

On aboutit ainsi à l'algorithme suivant.

Algorithme Programmation dynamique

1. Marquer les nœuds terminaux avec leur valeur donnée.
2. En tout nœud dont tous les nœuds immédiatement aval sont déjà marqués, faire :

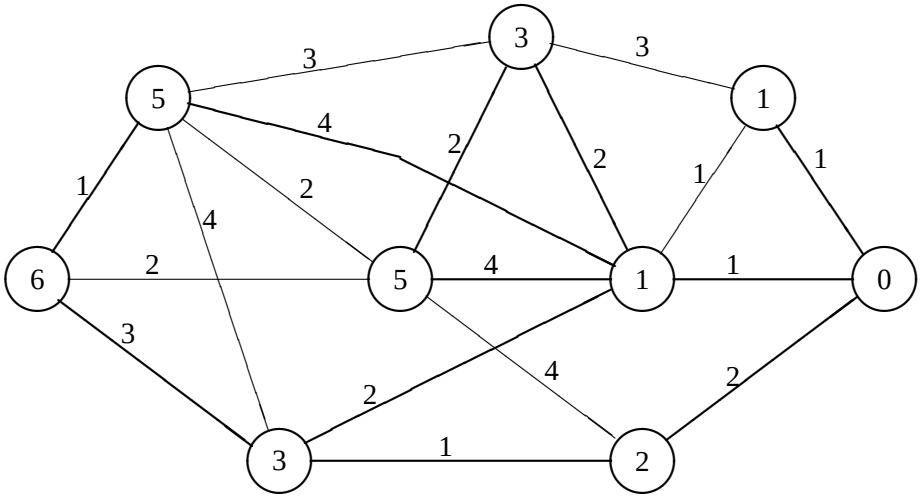

FIG. 4.2 – Le graphe de la figure 1 avec les valeurs et les arcs optimaux

- pour chaque nœud immédiatement aval, calculer la somme de la longueur de l'arc vers ce nœud et de la valeur de ce nœud.
- Prendre la plus petite de ces sommes pour valeur du nœud courant, et le marquer.
- Marquer le, ou les, arc(s) donnant la valeur retenue.

3. Retourner en (2) jusquà ce que tous les nœuds soient marqués
4. Tout chemin partant d'un nœud initial de valeur minimum et n'empruntant que des arcs marqués est optimal, et a une longueur égale à la valeur marquée à ce nœud.

Dans l'exemple de la figure 3, qui a deux nœuds initiaux possibles et trois nœuds terminaux, on a seulement attaché des poids aux nœuds terminaux, puisque des poids sur les nœuds intermédiaires auraient seulement augmenté d'autant le poids de chaque arc les quittant, sans augmenter la généralité de l'exemple.

On donne directement le graphe marqué avec les valeurs et les chemins optimaux. On conseille au lecteur de refaire l'algorithme lui-même pour constater combien il est simple et rapide. Pourtant ce graphe a plus de 110 chemins possibles.

De très nombreux problèmes combinatoires peuvent se ramener à un problème de recherche de chemin de poids minimal dans un graphe. La caractéristique essentielle pour mettre à jour une telle structure, et l'exploiter, est le sens de parcours unique. En général il provient de ce que le problème peut être organisé en "étapes" dont l'ordre est imposé par la nature du problème, ou immatériel quant à la solution cherhée de sorte qu'il peut être fixé arbitrairement. Cet aspect d'étapes successives, ou dynamique, va être examiné maintenant.

4.2.2 Système dynamique et programmation dynamique

Système dynamique

Nous examinons maintenant un cas particulier extrêmement important. Supposons que les nœuds du graphe, appelés ici *états*, peuvent être repérés par un numéro d'étape $k \in 0, 1, \ldots, N$, et pour chaque étape k soit X_k l'ensemble des états possibles à cette étape. L'hypothèse ici est que les arcs

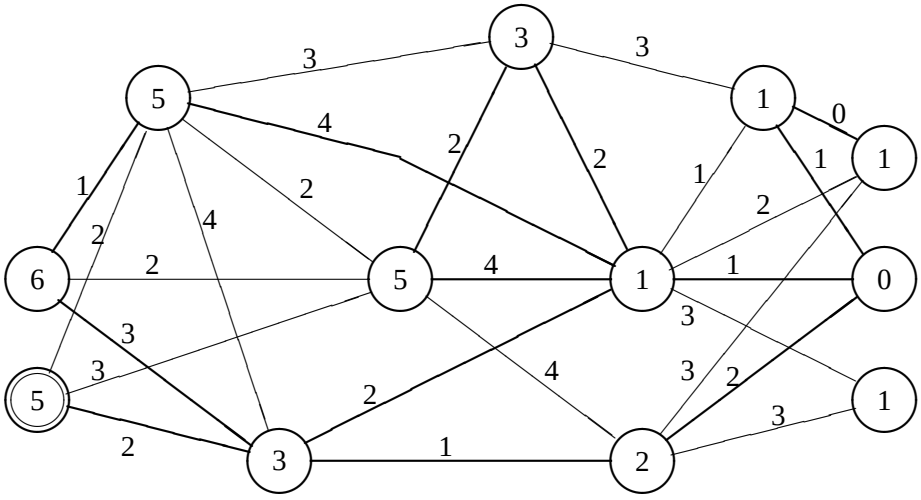

FIG. 4.3 – Un graphe à plusieurs nœuds initiaux et terminaux

relient toujours un état d'une étape à un de l'étape suivante, et que ceci correspond au sens de parcours imposé du graphe. Indiçons les arcs issus d'un nœud $(k, x(k))$ —où $x(k) \in \mathsf{X}_k$ est un nœud de l'étape k— par un indice $u \in \mathsf{U}(k, x(k))$ appelé *commande*. $\mathsf{U}(k, x(k))$ est simplement un ensemble dont le cardinal est égal au nombre d'arcs quittant le nœud $(k, x(k))$. On voit que le graphe définit une *équation dynamique* de la forme

$$x(k+1) = f(k, x(k), u(k)) \tag{4.6}$$

puisque pour chaque nœud $(k, x(k))$ et chaque commande $u \in \mathsf{U}(k, x(k))$, il définit à quel nœud ou état conduit cet arc, état toujours situé à l'étape $k+1$.

Un chemin dans le graphe est une suite d'états $\{x(k), k = 0, \ldots, N\}$, appelée *trajectoire*. Une trajectoire peut aussi être caractérisée par l'état initial $x(0)$ et une suite de commandes $\{u(k), k = 0, \ldots, N-1\}$, qui, via l'équation (4.6), engendre une trajectoire unique.

Notons encore $L(k, x, u)$ le "poids" ou la longueur —nous dirons ici le *coût*— de l'arc issu du nœud (k, x) indicé par u, et pour tout nœud terminal $x \in \mathsf{X}_N$, $K(x)$ le coût attaché à ce nœud. Ainsi, le coût d'une trajectoire, qu'il s'agit de minimiser, est donné par

$$J = K(x(N)) + \sum_{k=0}^{N-1} L(k, x(k), u(k)) \,. \tag{4.7}$$

Il y a équivalence complète entre un graphe structuré comme on l'a dit et une équation de la forme (4.6), et donc entre un problème de plus court chemin dans un tel graphe et le problème de minimisation de J donné par (4.7) avec la dynamique (4.6). Bien des problèmes seront formulés sous la forme (4.6),(4.7). Un bon moyen de les résoudre est de faire l'analogie avec le graphe, et de faire l'algorithme de programmation dynamique sur ce graphe.

Le terme même de "programmation dynamique" vient de là. L'équation (4.6) définit ce qu'on appelle un *système dynamique*. L'indice k est généralement interprété comme représentant le temps. L'équation (4.7) définit une fonctionnelle additive de la trajectoire. On recherche la commande, et éventuellement l'état initial, qui minimise ce critère ou coût.

Une forme équationnelle de la programmation dynamique

Ayant décrit le graphe et le critère par des équations, on peut décrire dans ce langage l'algorithme de la programmation dynamique. Notons $V(k,x)$ le "marquage" associé au nœud (k,x). On l'appelle plutot ici la fonction "performance", ou "de Bellman". L'algorithme que nous avons décrit s'écrit alors :

$$\forall k \in \{0,\dots,N-1\}\,, \forall x \in \mathsf{X}_k\,, \quad V(k,x) = \min_{u \in \mathsf{U}(k,x)} \left[L(k,x,u) + V(k+1, f(k,x,u))\right], \quad (4.8)$$

$$\forall x \in \mathsf{X}_N\,, \quad V(N,x) = K(x)\,. \quad (4.9)$$

L'algorithme de la programmation dynamique consiste donc à appliquer la formule (4.8) pour calculer V de proche en proche, en commençant par initialiser V avec (4.9), puis en reculant en k. Il faut avoir deux tableaux des valeurs de $V(k,\cdot)$ en mémoire, celui qu'on est en train de remplir et celui qui est utilisé dans le deuxième membre de (4.8). In fine, le tableau des $V(0,x)$ donne pour tout état initial possible le coût minimum possible.

Il faut aussi en même temps remplir un grand tableau des valeurs de u qui donnent le minimum à chaque (k,x). Ce tableau donne la *stratégie* (ou *commande en boucle fermée*) optimale, en ce que pour chaque état (k,x) il donne la commande optimale si on se trouve en cet état. Cet aspect est particulièrement utile si l'équation (4.6) constituait une approximation d'un phénomène physique, de sorte qu'on est susceptible de constater au cours de la mise en œuvre qu'on est dans un état $(k,x(k))$ différent de ce que à quoi on s'attendait. L'algorithme ci-dessus a donné une commande conseillée pour *tout* état dans le graphe. Certes, l'écart entre le phénomène réel et le modèle fait que cette commande n'est plus tout à fait optimale, mais si cet écart est faible, elle a toutes les chances de rester une "bonne" commande.

Système à état et continu

Tout le développement de la programmation dynamique a été fait en termes de graphe, avec donc l'hypothèse constante que dans l'équation (4.6), x et u prennent leurs valeurs dans des ensembles X_k et $\mathsf{U}(k,x)$ finis. Cependant, ces équations ainsi que (4.7) gardent un sens si ces ensembles sont des sous ensembles de $\mathbb{R}^n$ et $\mathbb{R}^m$, disons, respectivement. C'est à dire qu'alors l'état est constitué de n nombres réels et la commande de m nombres réels, les uns et les autres éventuellement bornés.

Les équations de la programmation dynamique (4.8)(4.9) gardent également un sens. Et on va démontrer le résultat suivant :

Théorème 4.7 *S'il existe une fonction rélle $V(\cdot,\cdot)$ satisfaisant les équations (4.8) et (4.9), en désignant par $\varphi(k,x)$ un argument du minimum dans (4.8), si la commande $u(k) = \varphi(k,x(k))$ est admissible pour x_0 (au sens où elle engendre une trajectoire qui respecte les contraintes $x(k) \in \mathsf{X}_k$, ce dont on peut s'assurer par un choix convenable des $\mathsf{U}(k,x)$), alors cette commande est optimale pour le problème défini par (4.6)(4.7) initialisé en $x(0) = x_0$.*

Démonstration Soit $\{u(0), u(1), \dots, u(N-1)\}$ une commande admissible, engendrant une trajectoire $\{x_0, x(1), \dots, x(N)\}$. En tout point de cette trajectoire, d'après (4.8), on a

$$V(k,x(k)) \leq L(k,x(k),u(k)) + V(k+1, f(k,x(k),u(k)))\,,$$

ou encore

$$V(k,x(k)) - V(k+1,x(k+1)) \leq L(k,x(k),u(k))\,.$$

Sommons cette inégalité de $k = 0$ à $N - 1$, il vient

$$V(0, x_0) - V(N, x(N)) \leq \sum_{k=0}^{N-1} L(k, x(k), u(k)) .$$

Utilisons alors (4.9) pour exprimer $V(N, x(N))$, que nous faisons repasser à droite, il reste

$$V(0, x_0) \leq K(x(N)) + \sum_{k=0}^{N-1} L(k, x(k), u(k)) ,$$

soit

$$V(0, x_0) \leq J(x_0, \{u\}) . \tag{4.10}$$

Maintenant, si la suite des $\{u(k)\}$ coincide pour tout k avec $\varphi(k, x(k))$, les inégalités dans les calculs ci-dessus sont toutes remplacées par des égalités, et on conclu que

$$V(0, x_0) = J(x_0, \varphi) . \tag{4.11}$$

La comparaison des relations (4.10) et (4.11) établit le théorème.

Système en temps continu

Dans la pratique, le système (4.6) est souvent issu de la discrétisation en temps d'un système en temps continu, ou système différentiel, de la forme

$$\dot{x} = F(t, x, u) .$$

Si on discrétise ce système avec un pas de temps h, en notant $x_k = x(kh)$ et $u_k = u(kh)$, on a au premier ordre

$$x_{k+1} = x_k + hF(kh, x_k, u_k)$$

qui est bien une équation de la forme (4.6), avec

$$f(k, x, u) = x + hF(kh, x, u) .$$

De même, le système différentiel peut être muni d'un critère intégral à minimiser, de la forme

$$J = K(x(T)) + \int_0^T l(t, x(t), u(t))\, dt$$

qui peut être approximé au premier ordre par une somme finie (où $T = Nh$)

$$J = K(x_N) + \sum_{k=0}^{N-1} hl(kh, x_k, u_k)$$

qui est bien de la forme (4.7).

Ainsi, la programmation dynamique apparait comme un moyen d'aborder l'optimisation d'un critère intégral pour un système différentiel, un problème connu sous le nom de "commande optimale", ou en Franglais de "contrôle".

L'approximation au premier ordre proposée ci-dessus n'est convenable que si on choisit un pas de temps "assez petit". De même, l'application pratique demandera souvent qu'on discrétise aussi x et u, se ramenant en fait à un problème fini. Et encore, cette discrétisation elle-même demande à être faite avec soin. Enfin, ces discrétisations ne mèneront à un problème faisable en pratique que si les dimensions des espaces d'état et de commandes sont assez petites pour que le nombre de "nœuds" du graphe soit raisonnable.

En fait, ce que nous obtenons ici est *une* discrétisation de l'équation de Hamilton Jacobi Bellman, l'équation aux dérivées partielles équivalente pour ce problème continu à l'équation de Bellman pour le problème à temps discret. Une analyse plus approfondie des schémas numériques nécessiterait beaucoup plus de mathématiques. Nous nous limitons donc à cette aproche naïve.

Mais quelles que soient les limitations de cette méthode, elle reste extrêmement utile dans des cas où elle s'applique. En particulier par le fait qu'elle se prête à prendre en compte toutes sortes de contraintes sur les états admissibles, et des données sans bonnes propriétés mathématiques. (Par exemple, les données peuvent dépendre de fonctions tabulées, etc.)